

Automatic Keyphrase Extraction on Amazon Reviews

Ruiqi Chen

University of Tampere
School of Information Sciences
M.Sc. thesis
Supervisor: Jyrki Nummenmaa
June 2018

University of Tampere
School of Information Sciences
Software Development
Ruiqi Chen: Automatic Keyphrase Extraction on Amazon Reviews
M.Sc. thesis, 60 pages, 3 index pages
June 2018

Abstract.

People are facing severe challenges posed by big data. As an important type of the online text, product reviews have evoked much research interest because of their commercial potential. This thesis takes Amazon camera reviews as the research focus and implements an automatic keyphrase extraction system. The system consists of three modules, including the Crawler module, the Extraction module, and the Web module. The Crawler module is responsible for capturing Amazon product reviews. The Web module is responsible for obtaining user input and displaying the final results. The Extraction module is the core processing module of the system, which analyzes product reviews according to the following sequence: (1) Pre-processing of review data, including removal of stop words and segmentation. (2) Candidate keyphrase extraction. Through the Spacy part-of speech tagger and Dependency parser, the dependency relationships of each review sentence are obtained, and then the feature and opinion words are extracted based on several predefined dependency rules. (3) Candidate keyphrase clustering. By using a Latent Dirichlet Allocation (LDA) model, the candidate keyphrases are clustered according to their topics. (4) Candidate keyphrase ranking. Two different algorithms, LDA-TFIDF and LDA-MT, are applied to rank the keyphrases in different clusters to get the representative keyphrases. The experimental results show that the system performs well in the task of keyphrase extraction.

Keywords: Review mining, Keyphrase extraction, Latent Dirichlet Allocation

Contents

1.Introduction.....	1
1.1 Background.....	1
1.2 Web Mining.....	2
1.3 Significance of Review Mining.....	4
1.4 Research Question.....	6
1.5 Research Task.....	7
1.6 Thesis structure.....	7
2. Literature Review on Review Mining.....	8
2.1 Procedure of Review Mining.....	8
2.1.1 Data Collection.....	8
2.1.2 Feature Extraction.....	9
2.1.3 Opinion Extraction.....	13
2.1.4 Clustering.....	14
2.1.5 Ranking.....	17
2.2 Overview of Review Analysis Systems.....	19
3. Amazon Keyphrase Extraction Process.....	21
3.1 Data Collection.....	21
3.1.1 Web Crawler.....	21
3.1.2 Amazon Review Characteristics.....	22
3.2 Feature and Opinion Extraction.....	23
3.2.1 Preprocessing.....	24
3.2.1.1 Spam Detection.....	24
3.2.1.2 Lemmatization.....	25
3.2.1.3 Cleaning.....	25
3.2.1.4 Segmentation.....	25
3.2.2 Pattern Extraction.....	25
3.2.2.1 Semantic Analysis.....	26
3.2.2.2 Feature and Opinion Extraction.....	26
3.2.2.3 Pruning.....	29
3.3 LDA-Based Keyphrase Clustering and Ranking.....	29
3.3.1 LSI and PLSA.....	30
3.3.2 LDA.....	33
3.3.3 Clustering Keyphrases Based on LDA Model.....	35
3.3.4 Keyphrase Ranking.....	37
3.3.4.1 LDA-TFIDF.....	38
3.3.4.2 LDA Max Topic(LDA-MT).....	40

4. Evaluation	41
4.1 Data Set	41
4.2 Evaluation Criteria	41
4.3 Result Analysis	42
5. Implementation of the Keyphrase Extraction System	45
5.1 Crawler Module	45
5.2 Extraction Module	47
5.3 Web Interface Module	48
6. Conclusion	50
References	53

1.Introduction

1.1 Background

The volume of web content is increasing rapidly with the development of information technology. Nowadays people can create tremendous amounts of data every day in all kinds of forms, for example, news, articles, advertisements, and reviews. As it is known that humankind is entering into a new era of Big Data, it is not difficult to realize how data is changing our lives. For example, in 2009 Google successfully predicted the diffusion regions of the H1N1 virus a few weeks before it hit the headlines. They made several correlation models based on user search queries, and the results turned out to be even more timely than official announcements [1]. However, public health is not the only area where big data can make a difference: other industries like education and engineering are progressively focusing on the research of big data as well.

Big data is bringing significant convenience to people's work and life. Donald Trump posted a tweet on Twitter, and a second later millions of Twitter users knew what he said. Companies like Google provide people with an easier way to search for information on the internet. From push services of mobile applications, people can always acquire worldwide news immediately.

Although big data is changing the world, the massive amount of information that people are creating every day make it challenging to process manually. The first problem is the limit of processing speed. Some real-time data need to be processed directly; otherwise, it would be no longer useful. For example, in the case of stock data, a decision made on a piece of old stock data will probably lead to a huge loss. Besides, human resources are no longer sufficient for handling the growing amount of data. As reported by David Sayce [3], around 52 million tweets were produced every day in 2016. Such a volume of data is impossible for people to process. Moreover, although the amount of data is getting larger, valuable data only occupies a small fraction of the whole, hidden among other useless data.

Data mining was born to solve the above-mentioned problems. Data mining aims to quickly find the potential knowledge and possible correlations inside a data source, which can facilitate people to solve problems in different fields.

One crucial area of big data, online product reviews, requires more attention as “90% of consumers read online reviews before visiting a business, and 88% of consumers trust online reviews as much as personal recommendations” [4]. Currently there are quite many e-commerce companies around the world. Amazon, which started as an online bookstore in 1995, has now become one of the most popular e-commerce companies in the world providing an extensive range of goods. In 2014, Amazon received 334,605 reviews per month [5]. Such a large number of reviews provides a good reason for review analysis.

1.2 Web Mining

With the rapid growth of the web, an increasing amount of web data can be easily accessed. There are more than 1 billion websites in the world wide web and the number is still growing [6]. Unlike traditional resource formats such as books or expert advice, web data is much easier to acquire and utilize. Seeing the potential value of web data, many researchers have started to focus on web mining.

Nowadays, the problem that web data analysis faces is with too much available data rather than too little: the volume of web data and the widely scattered locations of the data is what makes comprehensive analysis difficult. However, web mining techniques can be used to classify web documents, extract document topics, and analyze user behaviour. With the help of web mining techniques the users can gather information comfortably and efficiently. In addition, web mining can help to optimize website structure and to provide personalized service based on user behavior.

Web mining aims to discover and extract valuable information from various web data [7]. However, different from general data mining which is usually applied on database data, web data mining needs to deal with more unstructured data such as text in natural language, which is difficult for a machine to understand [8]. Besides, web data has a lot of noise. A lot of noise meaning content unrelated to the intended goal of the data mining. That noise can be an advertisement, an irrelevant information in header and footer, or even user-generated data unrelated to the data mining goal, which can all complicate web mining.

There are three different types of web data: web hyperlinks, page contents, and usage logs. Accordingly, web data mining can be divided into three subtasks: web structure mining, web usage mining and web content mining [7].

Web structure commonly refers to web hyperlink structure [7]. A typical web page contains not only the text content, but also many page links. When trying to estimate the importance of each web page, the information hidden behind these links can be utilized to increase the estimated importance of the web pages. The basic concept is that, a web page that is more referred to by other web pages has more importance. Therefore, mining web structure information becomes very useful. Google's Pagerank is one of the most famous techniques in this area. It can rank subparts of large web pages based on their internal hyperlink structures, providing accurate searching results to users from all over the world.

Web usage data refers to the records of user interaction with a web site [7]. By studying user access logs, the system will learn the interests and habits of the individual user to predict future user action. Besides the server side log, the client side log, transaction information, and cookies can also be useful for mining. A recommendation system is a typical application which utilizes the user browsing history to estimate user preference.

Web content includes text, audio, image and anything that can be displayed on a web page [7]. Web text has various types of content with a corresponding variety of layout and formatting, such as news, reviews, articles, and blogs. The massive amount of web text is a substantial mineable resource, in fact, a lot of studies [9, 10, 11, 12] have been done on this area in the past twenty years.

The general tasks of web text mining include, for example, summarizing, classifying, clustering and association analysis, the results of which can be further utilized to develop higher level systems. For example, Andranik Tumasjan [13] used LIWC, a text analysis software on the Twitter corpus and proved that Twitter is a valuable resource for predicting election results.

A common task of web image mining is image retrieval, that is, to detect a user-intended image from tremendous web image resource based on user input. An early study [14] developed an image search system in which the user needed to select several example images and then the system returned the most similar image results. A recent study [15] reported a higher precision in image retrieval based on a 2D affine transformation between the user query and candidate images. For audio mining, an important task is the 'query-by-humming' proposed by Asif Ghias [16] in 1995.

Nowadays, this problem has well-performing solutions that have been applied to many music applications such as Midomi¹ and Shazam².

As an essential part of web content, online product reviews have received a lot of research attention recently. Meanwhile, the phrase “review mining” has become widely used. Review mining, which is also called opinion mining, aims to extract critical consumer opinion towards a product from massive unstructured review texts. In this respect, researchers such as Hu and Liu [9] as well as Pang and Lee [17] have made significant contributions to review mining. This thesis will also focus on the analysis of product reviews. The next section will talk about the significance of review mining.

1.3 Significance of Review Mining

A problem of offline shopping is that many companies use advertising or hire salespeople in order to attract customers. However, customers are easily misled by these advertisements because they may not reflect the true quality of the product. On the other hand, some well-performing products that are not prominently advertised can also be easily ignored by customers. Another problem is that companies tend to produce several similar products trying to satisfy different types of customers, thus it is more difficult to judge whether a product meets one’s specific needs. In addition, some products may have drawbacks that can only be realized after using it for a short period, increasing the risk of making a purchase without any advice.

However, e-commerce as a flourishing industry is changing the way that people are used to live. Different from traditional shopping mode, online shopping greatly enhances the information exchanging among consumers, and it allows consumers to shop anywhere at anytime. One of the most famous e-commerce companies in the world is Amazon.

Figure 1 is a typical camera product page of Amazon and Figure 2 shows some of the reviews of the camera. As shown in the figures, when browsing the web page, people can quickly acquire both the detailed information of the product and the experience of other consumers. With the popularization of e-commerce, today lots of consumers prefer to check reviews online before they decide to buy a product. As reported by

¹ <https://www.midomi.com/>

² <https://www.shazam.com/>

Khalid Saleh [18], 90% of the consumers read online reviews before making a decision.

The significance of analyzing product reviews has become increasingly apparent, which can be seen from two aspects. From the consumer's perspective, the comments from other consumers are very important and valuable, because mostly those comments contain the user experience of the product, which can be taken as quite good advice to support decision making. From the manufacturer's perspective, the reviews also reveal the quality of the product. By gathering the consumers' reviews the manufacturer will know how to improve the product quality, thus increasing the sales and gaining more profit. Besides, by analyzing reviews in different time periods, the manufacturer can get a good vision of the market trends, helping them to make a good self-positioning.

Figure 2 shows that the camera has over 1000 reviews, and there are several such products. Facing such a number of reviews, reading them one by one is impossible. Therefore, it is necessary to apply data mining techniques on the reviews and to transfer the unstructured texts into organized knowledge, making it easier for people to catch the key information.

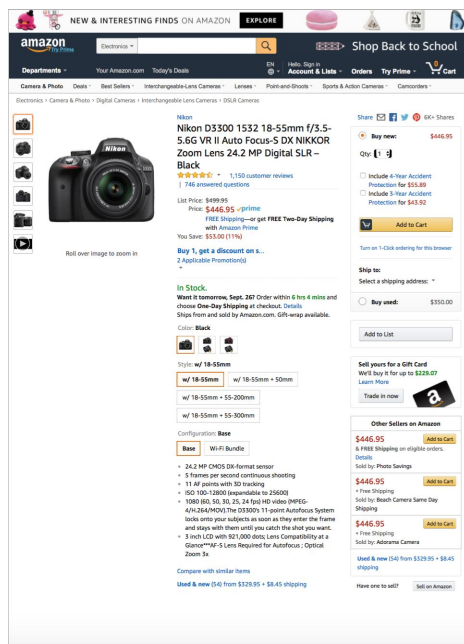


Figure 1. Camera product page

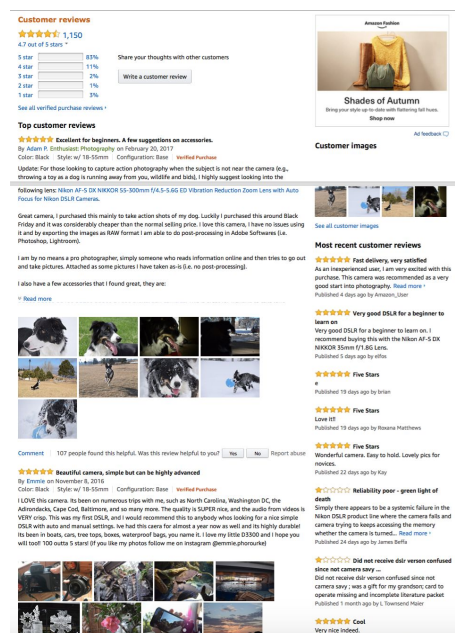


Figure 2. Camera product reviews

An online product review has two kinds of data: review texts and ratings. Some websites such as Thinkgeek³ only support text reviews, while more of the sites support both text review and rating, such as Amazon, BestBuy⁴, and AOSO⁵. The combination of text and rating make the reviews more explicit so that customers can quickly get an impression of the product. However, different persons have different opinions. As for products, customers may have different requirements on different product aspects. Taking mobile phones as an example, some customers are in favor of a large screen, whereas others may think a large screen as a defect because it cannot fit in the pocket. The overall rating cannot describe a product very well in detail. Hence, instead of overall rating, a fine-grained method is needed to extract the information accurately and precisely.

Given the above, this thesis takes product reviews as the research target, conducts a study around the collection of review data, the extraction of the opinions and the ranking of the results. This thesis aims to develop a robust system which can assist people in their decision making as well as reveal the potential improvements in product design.

1.4 Research Question

The purpose of this thesis is to develop an effective system to analyze product reviews. Due to their large quantity and reasonable quality, this thesis takes Amazon reviews as the data domain. Also, Amazon review pages are clearly structured, which makes them easier to crawl and analyze. First, a collection of Amazon product reviews will be crawled from the internet and stored into a local database. Then, some relevant techniques will be applied to the review data and the expected output is a list of keyphrases. Finally, the results will be compared and evaluated. Therefore, the research aims to answer the following questions:

1. How to define appropriate patterns to extract the candidate keyphrases from product reviews?
2. How to select representative keyphrases and make sure they are semantically different?

³ <https://www.thinkgeek.com/>

⁴ <https://www.bestbuy.com/>

⁵ <https://www.asos.com/>

1.5 Research Task

This thesis focuses on the mining of Amazon reviews. Given all the reviews of a single product, the proposed algorithm is expected to summarize the reviews into several keyphrases. These keyphrases should consist of nouns and adjectives or nouns and verbs in the order of importance. Adverbs are optional. To achieve the goal, several natural language processing (NLP) techniques will be applied to the review text. Statistical characteristics of the words are often utilized when extracting feature words and opinion words. Statistical characteristics include TF (Term Frequency), IDF (Inverse Document Frequency), first occurrence and length. Such features are easy to acquire. However, they have some limitations in more complex tasks. Thus, semantic information of the word is also needed to overcome the problem, which normally includes POS (part-of-speech), synonym and dependency relations.

In this thesis, a review analysis system will be developed to summarize the product reviews. In the system, keyphrases are extracted by several dependency rules. Spacy⁶ will be used to parse the text and reveal the dependency relations of review sentences. After getting the candidates set, latent Dirichlet allocation (LDA) will be employed to cluster the candidate keyphrases to ensure the results are semantically independent. For each cluster, the system will calculate the score of each keyphrase using LDA-TFIDF and LDA-MT separately. Keyphrases with the highest score will be selected as representative tags for the product. In this thesis, reviews from two camera products will be analyzed in the experiment, include Kodak PIXPRO AZ251 and Sony Cyber-Shot DSC-RX100.

1.6 Thesis structure

The thesis will answer the above questions in the following chapters. Chapter 2 presents a literature review on the area of review mining is presented. Chapter 3 introduces the methods and techniques that are employed in the process of keyphrase extraction. Chapter 4 performs a detailed evaluation of the proposed system. Chapter 5 mainly introduces the design and implementation of the keyphrase extraction system. Chapter 6 summarizes the thesis, and suggests some potential improvements in future work.

⁶ <https://spacy.io>

2. Literature Review on Review Mining

This chapter reviews some relevant studies on review mining. According to the general procedure of review mining, this chapter summarizes the relevant researches for each step. In addition, several famous review mining systems are introduced at the end of this chapter.

2.1 Procedure of Review Mining

Similar to data mining, review mining has several sub-tasks. Popescu and Etzioni [19] define four general steps of review mining: 1) Extract product features. 2) Identify opinion words related to features. 3) Calculate polarity of opinions. 4) Summarize and rank the results.

Figure 3 describes the processes in a flow chart. However, in this thesis, steps 1,2 and 4 are mainly focused on because the purpose of this thesis is to identify keyphrases which can summarize the reviews. However, the polarity information will also be involved in the results. In addition, a brief literature review of data collection will also be performed.

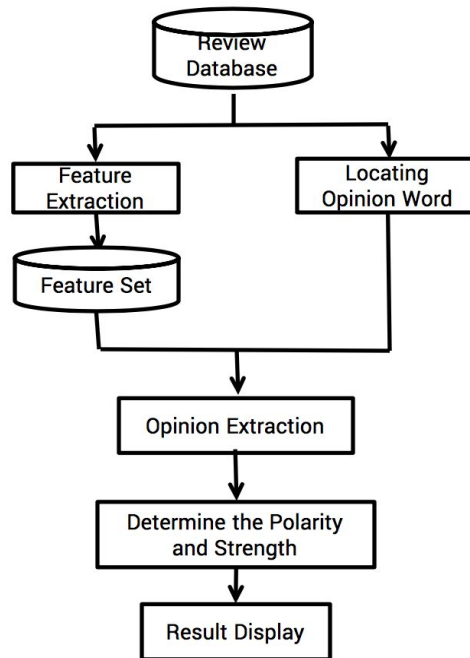


Figure 3. A framework of product review mining, interpreted from the description of Popescu and Etzioni [19] into a figure by the author of this thesis.

2.1.1 Data Collection

As the first step of data mining, data collection is always a crucial and necessary procedure. Nowadays, there are plenty of public datasets online for research use, such

as SNAP [20], a colossal Amazon review dataset including around 35 million reviews, and OpinRank [21], which contains cars and hotels reviews collected from Tripadvisor⁷ and Edmunds⁸. Such datasets can be acquired quickly, eliminating the need to obtain data separately. For example, Ling et al. [22] use SNAP dataset to develop their recommender system, and Zhang et al. [23] use Yelp dataset in their experiment.

However, most of these datasets are lacking maintenance and update, which means the data in them might be out of date. Therefore, more researchers choose to collect their own data to ensure the data quality.

In general, the collecting of the data is done by a system called web crawler. For single-format and straightforward data, the crawler can be very lightweight. Hu and Liu [9], Kasper and Vela [24], and Owsley and Sood [25] collected the reviews by a customized web crawler, and then they stored the data in a local database. For large scale, multi-format data, a more comprehensive and sophisticated crawler has to be developed. In this respect, Myllymäki [26] developed an XML based system ANDES, which can crawl relevant websites through a seed website and then extract domain-specific content from massive HTML structures. Chau and Pandit [27] proposed a parallel mining model, in their model a central server controls the mining task queue and assigns the tasks to different agents. The agent then executes the function in multithreading. They tested their model on an online auction website and greatly reduced the processing time. Similarly, Cheng [28] divided the large scale data mining task into several small jobs, and then ran them in parallel on different servers to improve efficiency.

Since the data of this thesis is only a small amount of Amazon product reviews, a customized web crawler is enough to accomplish the task. In Chapter 3, detailed information on the review collection process will be presented.

2.1.2 Feature Extraction

In most e-commerce websites, the product page often contains a short product description from the manufacturer. However, this kind of explanation is not a suitable resource for review mining, although it may involve information about product features. The reason is, manufacturers may have different concerns about product features from the consumers. Some electronic manufacturers like to provide

⁷ <https://www.tripadvisor.com>

⁸ <https://www.edmunds.com/>

information on technical details. For example, mobile phone manufacturers will probably focus on describing the clock speed of the processor, while most of the consumers are more concerned about the running speed when having a lot of applications installed. In addition, the manufacturer's description of the product is not complete. Some product features mentioned in user reviews are not taken into account by the manufacturer. Thus extracting the features from reviews is indeed necessary.

Product feature extraction is a crucial process of review mining. It aims to extract the product aspects which the consumers made comments on. Features are usually in the forms of nouns or noun phrases. Yi and Niblack [29] believe a feature must meet one of the following three conditions: 1) It has to be a part of the given subject. 2) It has to be an attribute of the given subject. 3) It has to be an attribute of a part of the given subject. Taking mobile phones as an example, the screen is a product feature; it is a part of the phone. The price is also a feature; it is an attribute of the phone. The image quality is a feature; it is an attribute of the phone camera, which is a part of the phone.

Product features can be divided into explicit features and implicit features [9]. As their names suggest, explicit features refer to the features that are explicitly mentioned in a sentence, while implicit features refers to the features that are not directly mentioned in a sentence. Implicit features can only be recognized after a deep-level understanding of the text. The following two review sentences are extracted from Amazon:

“I LOVE this camera - easy operation, great pictures. fantastic price. ”

“It's small enough to throw in my purse and easy to use.”

In the first sentence, it is easy to know that words “operation”, “pictures” and “price” are explicit features. In the second sentence, there is no such noun or noun phrase that could be taken as a feature. Only after understanding the whole sentence, it can be inferred that the author is talking about the size of the camera.

There are two ways of extracting explicit features, which are the manual definition and automatic extraction. The manual definition is to set up a feature vocabulary for products from a specific area. In the respect, Zhuang et al. [30] defined several classes (screenplay, character design, vision effects, actor and actress, etc.) for movie features by observing the reviews from IMDB, and then used a statistical method to determine the movie feature set.

Blair et al. [31] used a combinational approach of manual definition and automatic extraction to extract the features from local service reviews. They defined four features (food, decor, service, value) for restaurants and five features (rooms, location, dining, service, value) for hotels. For each feature set, they merged them with auto-extracted features to improve the overall accuracy of feature extraction.

Yao et al. [32] developed a supervised review mining system for automobiles based on a manually created ontology base. Their system comprehensively analyzed the opinions towards different features of a single car as well as a single feature from different cars.

Kobayashi et al. [33] also developed a semi-automatic system for collecting opinion expressions from game and automobile reviews. Given three manually selected seed sets of subjects (products), attributes (features) and values (opinions), their system can extract the evaluative expressions based on predefined co-occurrence patterns. However, a human judge is still needed to evaluate the expressions in the final step.

However, there are some drawbacks in the manual definition of the product feature. Firstly, with the rapid growth of the world economy, the variety of products is also increasing quickly, which means manual definition becomes especially unrealistic to cover all the product categories. Secondly, the manufactures often need to update their product design according to market research, while the manually defined features remain outdated, leading to inaccurate results of the system. Meanwhile, different domain experts are needed to create domain-specific features, which brings a considerable cost of time and money.

Automatic product feature extraction mainly employs the natural language processing techniques such as part-of-speech tagging, syntactic analysis and document pattern of words. Given a sentence, automatic feature extraction can locate the feature words based on some restrictions and predefined rules. Both supervised approach and unsupervised approach can be used to accomplish the task.

For supervised learning, Hu and Liu [34] manually labeled feature words that occur in the reviews. For convenience they separated the sentence into 3-gram segments and saved the segments in a transaction file. They then applied association rule mining [35] on the file to acquire common patterns, which can be used to identify possible features in new reviews.

Kessler et al. [36] focused on finding the semantic relationships between feature words and opinion words. They annotated both features and opinions in a dataset of car and digital camera reviews. Supervised machine learning was employed to rank possible features linked to an opinion word. Their algorithm yields a precision of 0.748 and recall of 0.654, and both are higher than the baseline algorithm, which was proposed by Bloom et al. [37] in 2007. Supervised approaches usually perform well on review mining, yet one disadvantage is the need for manual labeling in advance.

For unsupervised learning, Hu and Liu [9] applied POS tagging on review sentences and saved the noun/noun phrase in a transaction file. An association miner [38] was again used on the file to extract frequent features. Compactness pruning and redundancy pruning were also used to filter the result. The system can also identify infrequent features by checking if opinion words exist in the same sentence. Their system can extract the features from multi-domain reviews.

Kim and Hovy [12] employed a semantic role labeling approach to extract the topic (feature) and opinion holder from a sentence. Firstly, opinion words were extracted from the sentence, and a frame class was assigned to the sentence based on FrameNet data. They then labeled the sentence fragments with their semantic roles using a statistical method. A mapping between the semantic roles with opinion holder and topic (feature) was created manually to identify the feature and holder of the given opinion word. Their system yields an average precision of 0.618 on topic (feature) extraction, which is much higher than the baseline, which yields only 0.179. However, their system depends a lot on the external corpus, causing a risk of unstableness in future development.

On top of the Know-it-all system [39], Popescu et al. [19] developed an unsupervised review mining system called OPINE. Given an input of product class and predefined rule templates, the system can extract candidate features based on the rules. To improve the extraction accuracy, PMI (Point-wise Mutual Information) score, which depends on the hit counts from web searching, is calculated for each of the candidates to check the probability of it being a feature of the given product class. Their system receives a 22% higher precision over Hu and Liu's algorithm on the same dataset, while only has a 3% lower recall. However, since calculating PMI will consume a lot of time, their system is not suitable for large dataset mining.

Implicit features do not show explicitly in the sentence and are difficult to extract by machine. One concept to extract implicit features is to take it as a follow-up task of explicit feature extraction, which was used by Hu and Liu [34] in their system. In the

training set, if there is no feature word, they tag the opinion word and create a mapping between the opinion word to an assumptive feature word. By checking the mapping, the system can detect implicit features in new data. However, this approach needs human intervention and is difficult to adapt to a new domain.

Similarly, Hai et al. [40] used a co-occurrence association rule mining to find implicit features. First, they collected opinion words with corresponding explicit features and tried to find rules between them. Then for those opinion words without any feature words they used the rules to assign the most suitable feature words to them.

Qiu et al. [41] proposed a topic modeling based implicit feature extraction method. They regard product features as topics, and each word under one topic is a feature-related opinion word. In their concept, opinion words are not restricted to adjectives, but can be nouns or verbs too. However, implicit feature extraction still faces a lot of challenges. This thesis will only extract explicit features with related opinions.

2.1.3 Opinion Extraction

Opinion word refers to the word which the author uses to express her/his feeling about a product feature. Some researchers extract opinion words utilizing an opinion words dictionary. For example, Zhuang et al. [30] selected top 100 positive words and negative words with the highest frequency from their labeled training data and took these opinion words as the seed set. To find unobserved opinion words in training data, they iterated through WordNet⁹ and found the words with at least one seed word existing in their synsets, and then added these words into final opinion words list. Finally, they extracted the opinion words based on the opinion words list. Ku et al. [42] tried to create a Chinese opinion words dictionary for news and blogs. They first collected the opinion words from GI (General Inquirer)¹⁰ and CNSD (Chinese Network Sentiment Dictionary)¹¹ and then took these words as the seed set. They then expanded the opinion words by searching for their synonyms in CiLin (TongYiCiLin) [43] and BOW (Academia Sinica Bilingual Ontological Wordnet)¹². Lastly, they calculated a polarity score for each word based on a positive formula and a negative formula.

⁹ <https://wordnet.princeton.edu/>

¹⁰ <http://www.wjh.harvard.edu/~inquirer/>

¹¹ http://134.208.10.186/WBB/EMOTION_KEYWORD/Atx_emptwordP.htm

¹² <http://bow.sinica.edu.tw/>

Another approach to opinion word extraction is to discover the relations between feature words and opinion words. By observing the reviews, Hu and Liu [9] find that opinion words usually occur near to the feature word. According to this observation, they collected the opinion words by checking if adjectives exist near the feature word. For example, in the review “The appearance of this phone is good.”, they first locate the feature word “appearance” and then find the nearest adjective “good”. This approach is easy to implement, however, it only considers adjectives as opinion words, ignoring that some verbs and adverbs can also express the author’s attitude. For example, in the review “I love this phone.”, the word “love” indicates the semantic orientation too.

Inspired by Hu and Liu’s work, Popescu et al. [19] manually defined ten dependency relations between feature words and opinion words based on the parsed result from the MINIPAR¹³ parser. Their algorithm can detect not only adjective opinion words but also the noun and verb opinion words. However, opinion words that do not meet the rules will not be detected from the reviews.

In another paper proposed by Hu and Liu [44], they focused on analyzing the reviews in the form of “pros” and “cons”. Such kind of reviews commonly occur in the Amazon website. They developed a supervised method to mine the CSRs (Class Sequence Rules) from labeled reviews. The rules can then be used to identify feature words and opinion words in reviews.

Feng et al. [45] extracted the feature-opinion pairs based on some dependency relation rules. They first parsed the review text using Stanford Dependency Parser¹⁴, and then extracted the word pairs with three common dependency relations, including adjectival modifier (amod), nominal subject (nsubj) and direct object (dobj). Likewise, their algorithm can also detect verb opinion words.

Yi et al. [46] designed a system for review mining, which is called SA (Sentiment Analyzer). SA first extract feature words from review sentences, and then obtains the ternary expressions in the form of <target, verb, source> as well as binary expressions in the form of <adjective, target>. By using several external sentiment lexicons, the system can calculate the polarity of each expression.

2.1.4 Clustering

Unlike most studies that have been made on product reviews, this thesis does not focus on sentiment analysis or polarity classification, but on exploring the central

¹³ <https://gate.ac.uk/releases/gate-7.0-build4195-ALL/doc/tao/splitch17.html>

¹⁴ <https://nlp.stanford.edu/software/stanford-dependencies.shtml>

ideas of the review. The advantage of doing this is that people can have a quick overview of the product, knowing what other consumers were concerned about and how the product aspects are viewed by most consumers.

Product reviews could have a lot of features such as shape, size, color, quality, and cost-effectiveness. Different consumers have different considerations for each feature. Therefore, reviews need to be automatically clustered and grouped into different categories to reflect more detailed aspect-level information of the product. For example, taking hotel reviews as an example, users' reviews of a hotel mainly focus on "price", "service", "comfort", "location", etc. The most efficient way to summarize the product is to put each review or review fragment into the corresponding categories according to its semantic information so that consumers can get faster access to useful information.

This process aims to ensure the final keyphrases cover more information and do not overlap. For review mining, an important observation is that when people comment on product aspects, they tend to use similar words [9]. Therefore, clustering the candidate phrases based on product aspects is reasonable. Opinions from different consumers will be clustered together if they comment on the same product aspect. A sorting process can then be made for each cluster to select the representative tags of the product.

The easiest way to cluster the features is through a simple string matching process. Miao et al. [47] grouped similar feature words by using domain knowledge. For example, they think "battery" should be grouped into "battery life", and "picture" should be grouped into "picture quality". Another approach [48] is to stem the words and to check if a feature is a subset of another feature. The disadvantage of this approach is that it can not classify different feature words that are semantically similar, such as "price" and "cost". To solve this problem, more advanced algorithms are needed.

When clustering keyphrases, a problem is that the keyphrases are relatively short, so there is relatively little information for estimating statistical characteristics from the keyphrases. However, one solution is to use external dictionary and knowledge base to expand the keyphrases vocabulary, in order to enrich the semantics of keyphrases and thus to improve the clustering accuracy.

Huang et al. [49] used Wikipedia to map keyphrases from the text to Wikipedia's anchors and took these anchors as the topics of the text. Then, they performed clustering on different texts based on their topics.

Similarly, Banerjee et al. [50] used Wikipedia to expand the semantics of short text. They used words and phrases from short text to construct the search criterias and queried the Wikipedia document library for the most eligible articles as a feature extension for the original short text.

Hu et al. [51] proposed a novel short text clustering framework, which can improve the accuracy of clustering by extracting the internal semantics of short text along with associating external knowledge base and using a three-layer hierarchy method to deal with the sparse problem. They also adopted a Wikipedia and WordNet combined method to reconstruct the short text feature space. Finally, they applied K-means and EM algorithm to test the framework, which yields a better accuracy than the baseline methods.

Petersen and Poon [52] studied the previous feature extension methods and found that using a large external knowledge base, such as Wikipedia and external dictionaries, will increase the difficulty of clustering as well as the time consumption. Instead, they chose domain-relevant texts as the background knowledge base according to the field of the text being processed, which largely reduces the quantity of resources needed compared to using Wikipedia.

In addition to introducing external semantic knowledge, it is more effective to extract the internal semantic knowledge behind the text. In recent years, many researchers have applied topic modeling to the field of opinion mining to extract the topics of reviews. This is because a product feature can be regarded as a specific topic of the review text [53]. Currently, some popular models for mining internal semantic knowledge are as follows: Latent Semantic Indexing (LSI) [54], Probabilistic Latent Semantic Analysis (PLSA) [55] and Latent Dirichlet Allocation (LDA) [56]. However, this thesis selects the LDA model to model the text. This is because the parameters of the LDA model are independent of the size of the corpus, so it is more suitable for large-scale text mining. The details of the LDA model will be introduced in Chapter 3.

Although traditional text mining methods have already gained extensive research, traditional text mining algorithms cannot model short texts well [57]. However, topic modeling has been widely used in NLP tasks since the beginning of this century. Research shows that text clustering based topic mining algorithms are able to extract the topics of reviews [58].

In this regard, Lu et al. [59] used a probabilistic topic model to carry out the task of short review mining. Based on the characteristics of product reviews and the PLSA model, they proposed the structured PLSA and unstructured PLSA model and added the predefined topics as prior knowledge, which improves the accuracy of the product

feature mining. The resulting topic clusters are more suitable as the basis of product review summarization. They tested their algorithm on customer reviews from eBay¹⁵, and the experiment results show that this probabilistic topic model based review mining method has better performance than the traditional supervised methods, and has a good effect for the subsequent task of review summarization.

Titov [60] uses a model for review data where the topic distribution of the whole review corpus is fixed, but the topic distributions of each document in the corpus differ. They proposed a two-layer review mining model named MG-LDA. Their model has good performance on clustering product features. For example, for hotel reviews, the model will classify “transportation” and “walk” into the category of “location”.

Jo et al. [61] assume that each sentence in a review contains a product feature and a sentiment associated with it. They proposed an ASUM (Aspect and Sentiment Unification Model) model, which can successfully obtain the product features and their corresponding sentiments, and does not need any manual annotation.

Guo et al. [62] proposed a universal feature mining model for product reviews named mLSA (multilevel latent semantic association). The model has two-layer LaSA (latent semantic association) structures, the first layer maps the words into the different product features, and the second layer classifies the product features according to the context. Similarly, their model does not require manual annotation.

Tu et al. [63] used a topic model to describe the review dataset, selecting the most representative topic words as candidate concept words. Then, the semantic relationships between conceptual words were extracted by using WordNet, and the semantic distance between conceptual words were computed. Finally, they generated concept classifications based on multi-level hierarchical clustering.

In this thesis, LDA is used to cluster similar keyphrases. The method takes the candidate keyphrases from the previous step as documents and establishes the LDA model. The LDA model will assign a probability distribution over topics for each candidate keyphrase, which can be used to cluster the keyphrases into different topics.

2.1.5 Ranking

After getting the clusters, it is necessary to sort the keyphrases by importance. A traditional way is to collect the review sentences that comment on the same product feature and then list all the product features by frequency [9]. However, this approach

¹⁵ <https://www.ebay.com/>

can only reveal the importance of product features, but cannot get the most important keyphrase under each product feature.

That is to say, the traditional method assumes that all the opinion sentences that describe the same product feature have equal importance, which is not the case in reality. The more appropriate way is to set different weights for each keyphrase in the same group. A keyphrase with a higher weight means more consumers tend to hold such kind of opinion towards the corresponding product feature. By ranking the keyphrases, it is possible to summarize the product with respect to several product parameters, which could help consumers check if they meet their expectation. Also, product designers may have strong interests in the sorted results, which can tell what most consumers care about. Meanwhile, by filtering out the keyphrases with a lower score, the accuracy of the results can be improved.

A lot of researchers have been focusing on keyword extraction and ranking. A very fundamental way to get keywords is to count the number of occurrences of each unique word in the document and take top k words with the highest frequency as keywords. Based on this concept, one of the most famous algorithms is TFIDF, proposed by Salton and Buckley [64] in 1988.

TF can reflect the capacity for an individual word to describe the documents, while IDF can reflect the capacity for an individual word to distinguish the documents. The concept of TFIDF is that when a word occurs many times in one document but seldom occurs in other documents means this word has a strong capacity to represent the current document. That is to say, a word with a higher TFIDF score will be more important. The drawback of TFIDF is also obvious, since it only uses the statistical information of words, ignoring the semantic information behind the document.

Rose et al. [65] developed a rapid automatic keyword extraction method for individual documents. They calculated the word weight based on the word degree as well as the word frequency. For multiple word expressions, they calculated the weights by summing the members' weights up. Their approach proved to be very efficient and universal.

Furthermore, graph-based keyword extraction also yields considerable success [66, 67, 68]. The basic concept is to regard the document as a word-based network. TextRank [66] is one of the most famous algorithms in this area. Inspired by PageRank, TextRank takes words as the nodes of the graph, by setting a fixed-size window and moving it over the document the algorithm checks if two words co-occurred in the window. If yes, then add an edge between these two words in the graph. The algorithm will output the score of each node in the graph.

However, the above methods can not solve the problem of this thesis properly. The proposed system aims to extract multiple semantically different keyphrases from the reviews to summarize the product. A keyphrase is defined to be in the form of <feature, modifier, opinion>. The methods mentioned above can only be used to extract keywords or adjacent keyword lists but not keyphrases, which do not meet the requirement. Also, the meanings of the results are likely to be overlapped.

Therefore, this thesis uses two algorithms, LDA-TFIDF and LDA-MT to sort the keyphrases. Since the keyphrases have already been clustered in advance, we can assume that the semantic meanings of keyphrases from different clusters do not overlap.

2.2 Overview of Review Analysis Systems

Product review mining is one of the most popular research topic of text analysis and has attracted the attention of many scholars. Due to the significant application value of review mining in real life, a lot of mining systems have been developed during recent years.

Dave et al. [69] developed a review mining system called “Review Seer”. Their system is trained by self-tagged review data, and can automatically extract the features and opinions from the reviews. The system also scores each product feature by a machine learning algorithm to classify the review sentences as positive or negative.

Gamon et al. [70] created a system called “Pulse” which can analyze car reviews. The system first crawls the car reviews from the internet and creates separate collections for different car models. The system also embeds a sentiment classifier and a keyword extractor to determine the polarity of car features and reviews.

Similarly, Hu and Liu [34] developed an “Opinion Observer” system. “Opinion Observer” is the first system to allow multi-product comparison and it also gives a visual representation of the results.

Some researchers also focus on large-scale text analysis. “Web Fountain” [29] is such a system which can process various resource from internet in parallel. Two types of miners complete its core functions, one is entity-level miners that work on a single document, and the other is corpus-level miners that are used to analyze the entire dataset statistically.

However, all these systems mentioned above are based on traditional review mining methods such as POS tagging, name entity recognition, etc. Moreover, most of them

separate feature and opinion extraction into two steps, which ignore the latent relationships between feature words and opinion words and thus may cause information loss.

3. Amazon Keyphrase Extraction Process

This chapter explains the method and detailed process of Amazon review mining. In addition, some related technologies will be introduced. The entire data processing process is shown in Figure 4.

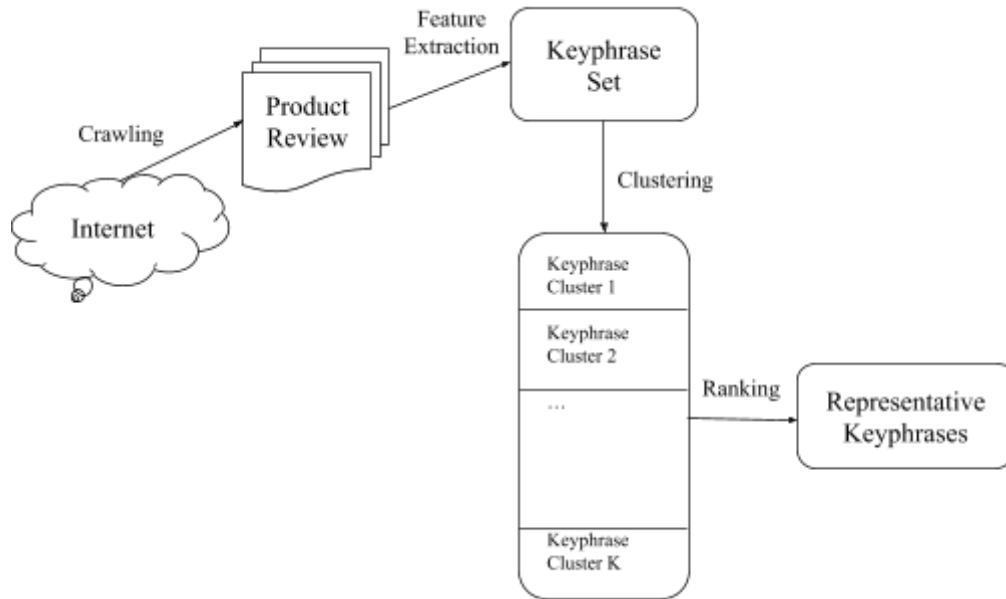


Figure 4. Processing flow of Amazon keyphrase extraction

3.1 Data Collection

3.1.1 Web Crawler

The first and essential step is to collect product reviews. In general, non-textual information on Web page like pictures, as well as HTML markup commands need to be removed when crawling. Currently, there are three common ways to get information on the webpage, including browser simulating based method, open API based method and web crawler based method [71].

Some browser-based plug-ins can simulate the browsers to crawl data. For example, Chrome widget CatGate¹⁶ can simulate Chrome to crawl reviews from Chinese social media WeiBo¹⁷. However, this kind of approach is complicated to implement and is not compatible with different browser kernel engines.

¹⁶https://chrome.google.com/webstore/detail/catgate/nncgefdjnpipajdfnindaiockdadpab?utm_source=www.cr4chome.com

¹⁷ <https://weibo.com/>

Using a website's open API to crawl reviews has become a popular approach recently. For example, TripAdvisor provides a public API platform for users to access their database. These APIs come with detailed documentation, making them straightforward to implement. However, the usage is subject to the limitations of the API provider, such as limitations on the number of visits, the accessing speed, and even the accessor IP. Unfortunately, Amazon does not provide such kind of open API for web users.

Therefore, this thesis uses a web crawler to crawl the reviews from Amazon. This approach is relatively simple to implement. In addition, it has high flexibility and less restrictions, which means more comprehensive content can be obtained on different demands.

The web crawler is used to obtain web data, which is an essential part of the search engine. The web crawler starts with a collection of seed URLs, then it gets the URL page and analyzes the page information. After that, it extracts useful contents and some new URLs, and puts the new URLs in the crawling waiting queue. It repeats the above processes until the crawl termination condition is met or the queue becomes empty.

This thesis uses Python to implement the crawler. Although Python is not as fast and stable as Java and C++, its grammar is simple to understand and there are a lot of mature external libraries to be used, which can greatly reduce the development difficulty.

3.1.2 Amazon Review Characteristics

Figure 5 shows a typical review on the Amazon website.

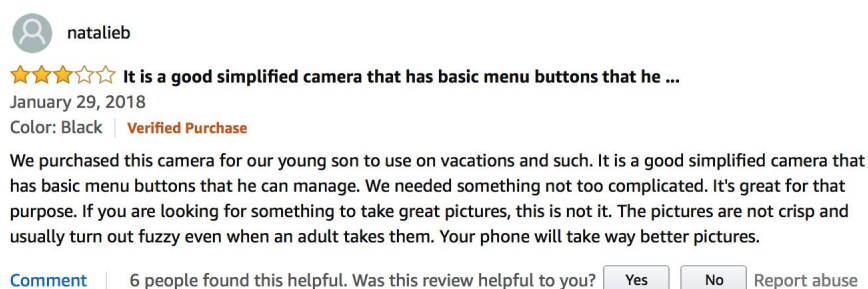


Figure 5. One example of Amazon review

It shows that an Amazon review usually consists of several parts, including username, rating, review title, review date, product color (optional), verified purchases, review content, and helpful vote. This information can be utilized in different review mining tasks.

For example, the username can be used to track the user's behavior. Some users like to review the product soon after they received it, and based on this, the recommender system can predict user's shopping tendency according to the purchase history. Review date can help manufacturers to perform market analysis. By tracking reviews at different periods, it is easy to discover the changes in the reviews, which is usually caused by a product update. The verified purchase label means the author purchased the product from Amazon, therefore it can help filter out false and spam reviews.

The review content part is the primary focus of this thesis. It contains a detailed evaluation of the product, making it very significant to be analyzed.

Concerning the length, a previous statistics [72] reports that the number of reviews that are 100-150 words is the largest, followed by the reviews of 150-200 words. However, the average amount of characters in a single review is about 582, which is a paragraph long. This also explains the importance of automatic review mining from another perspective, since exhaustively reading many reviews of such length will take a lot of time. In this thesis, the number of reviews that are 100-150 characters is also the largest. However, the average length of the reviews is a bit shorter.

Regarding the quality, Liu et al. [73] define an evaluation system SPEC for Amazon reviews, which divides reviews into four quality levels: “best review”, “good review”, “fair review” and “bad review”. The judging criteria include the number of evaluations on product features and also the clarity of evaluations. For example, “bad review” refers to reviews that do not evaluate any of the product features. On the contrary, reviews of higher quality-level have evaluated at least one product feature. They manually assessed 4909 Amazon's camera reviews, and the results show that 60% of the reviews are of “fair”, “good” and “best” quality. Their statistics show that the Amazon camera reviews have relatively good quality, hence mining Amazon camera reviews is significant.

Given a product URL, the Python crawler will crawl all the verified purchase reviews, including username, review date, review title, review content, and rating. All the reviews are then stored in the local database for subsequent processing.

3.2 Feature and Opinion Extraction

This section describes a sequence of steps to filter, lemmatize, clean, segment as well as perform feature and opinion extraction process for the reviews.

3.2.1 Preprocessing

After getting the reviews, the first thing that must be done is preprocessing. The reason for doing this is that raw reviews usually contain a lot of useless information such as symbols, numbers, etc, which could interfere with the later steps of the feature extraction process. In this thesis, the data preprocessing includes two steps: First, detecting and filtering the spam reviews; Then, apply basic preprocessing tasks on the reviews, including lemmatization, data cleaning, and segmentation. The entire preprocessing workflow is shown Figure 6.



Figure 6. Review preprocessing

3.2.1.1 Spam Detection

Spam denotes reviews that are irrelevant to the goal of the data mining as well as false reviews. The two reviews shown in Figure 7 are the examples of spam reviews. These two reviews are published for different product items from different users, but their contents are exactly the same. Such reviews are regarded as typical spam reviews.

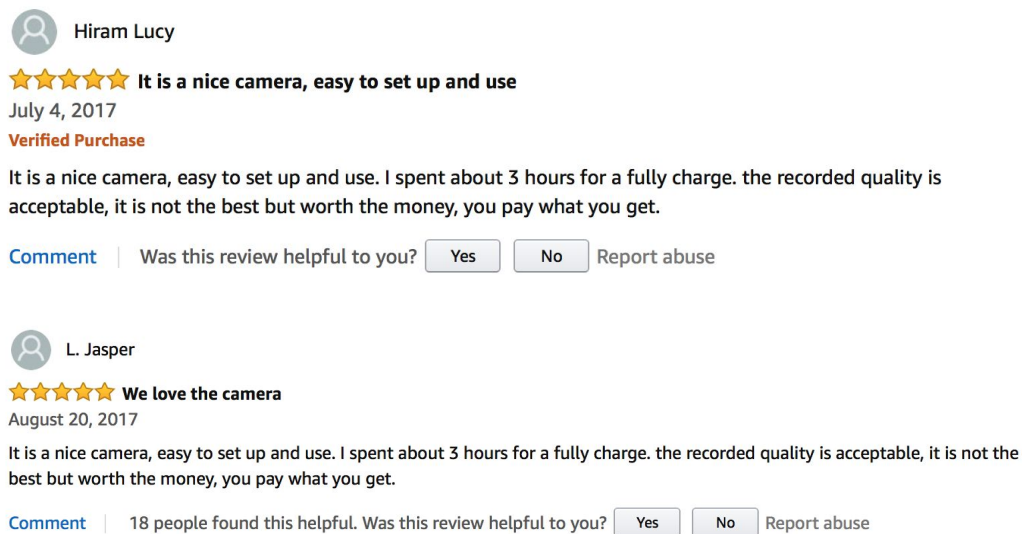


Figure 7. A typical example of spam review

There has been a lot of research on spam detection including detection of review spam. There are two kinds of detection methods that are commonly used: content based detection and reviewer behavior detection [74]. A typical characteristic of spam reviews is that they usually have high similarity. Taking advantage of this, one can detect spam reviews by calculating the similarity between two reviews. When the similarity exceeds a threshold, the reviews will be regarded as spam. The reviewer

behavior based method is to track the suspicious reviewers and treat all their reviews as spam. Suspicious behavior could be noticed by analysis of the review dates, the target objects, etc. However, this kind of method needs a lot of data support, which is not feasible in this thesis.

In this thesis, a similarity-based method is used to detect spam reviews. If two or more reviews have identical text contents, they are considered as spam reviews.

3.2.1.2 Lemmatization

Applying lemmatization on the reviews can greatly improve the extraction accuracy. After turning all words to lowercase, this thesis uses Spacy lemmatizer to lemmatize each word in the review text, transforming the original word into its basic form. For example:

is , are, was, were → be
phone, phones, phone's, phones' → phone

3.2.1.3 Cleaning

Cleaning mainly includes removing stop words and meaningless symbols. Stop words refer to those English words that do not have strong semantic content by themselves, such as 'the', 'is', 'that', 'at', 'which'. Removing these words does not have an impact on the text analysis in this thesis, as we do not perform grammatical analysis of long sentences where such words could be required. On the contrary, it helps to reduce the vocabulary size, which will improve the efficiency of analysis.

Secondly, online reviews like other informal texts could include symbols that we choose not to analyze in this thesis, such as emoticons, acronyms, and popular internet jargons. This thesis also carries on a filtration processing for these kinds of symbols.

3.2.1.4 Segmentation

Each review in the dataset needs to be split into sentences. This is because both the dependency relation analysis and the LDA model topic analysis in the subsequent process use sentences as the analysis unit. Therefore, this thesis splits the reviews by '.', '!', '?', ';', '\n', and then removes the sentences whose length are less than five characters.

3.2.2 Pattern Extraction

The product features include explicit features and implicit features as described in Section 2.1.2. However, implicit features are difficult to detect. Like other researchers [30, 9, 75], this thesis mainly studies the explicit features, which usually appear in the text in the form of nouns or noun phrases. Similar to Feng et al. [45], this thesis extracts features and opinions based on dependency relation rules. The advantage of

this method is that it can extract features and opinions at the same time. Besides, features and opinions of various parts of speech can be detected, which will greatly improve the semantic richness.

There are three steps in the pattern extraction process, as shown in Figure 8.

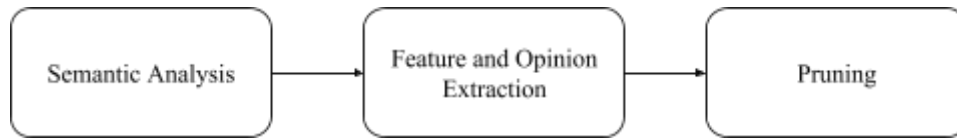


Figure 8. Pattern extraction process

3.2.2.1 Semantic Analysis

The semantic analysis includes POS tagging and dependency relation analysis. This thesis uses Spacy, an integrated natural language processing library, to carry out an in-depth semantic analysis of the text. The prepared review sentences are passed into the Spacy text analysis tool as input, and using the Spacy Pos tagger, the part of speech of each word is detected. Then, using the Spacy dependency parser, the dependency relations between words can be obtained. As shown in Figure 9, dependency relations connect pairs of words and relation is assigned a particular label describing the relation type, such as "nsubj" or "compound". In the figure, Arrows are dependency relations, dark gray words are relation types, and colored words at the bottom are part of speech labels.

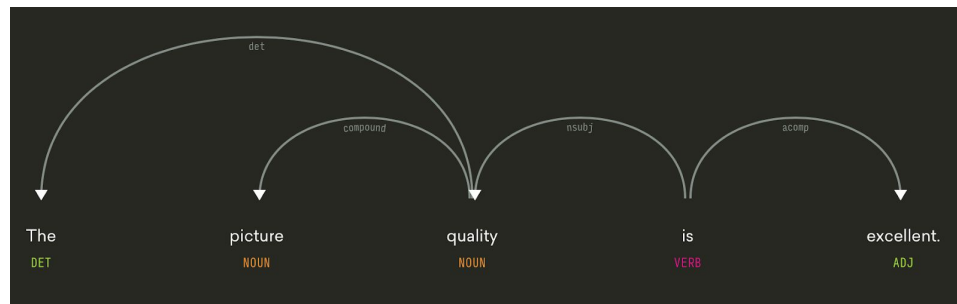


Figure 9. An example of dependency relation..

3.2.2.2 Feature and Opinion Extraction

By observing the dependencies from various reviews, some common dependency relations between feature words and opinion words can be found. These rules can be further used to catch feature-opinion pairs. This thesis builds three common rules in total. The point of using these rules is to discover these feature-opinion connections in complicated sentences where the feature and opinion words are not directly next to each other. First of all, the universal dependencies used in these rules are described in Table 1.

Universal Dependencies	Description
amod	Adjectival Modifier
advmod	Adverbial Modifier
acomp	Adjectival Complement
nsubj	Nominal Subject
neg	Negation Modifier
xcomp	Open Clausal Complement

Table 1. Description of universal dependencies

Three rules are as follows, where f represents the feature, o represents the opinion, and parentheses content is optional.

Rule 1: $\text{nsubj}_{\text{verb} \rightarrow \text{f}} + (\text{neg}) + (\text{advmod}) + \text{acomp}_{\text{verb} \rightarrow \text{o}}$

One example of this pattern is:

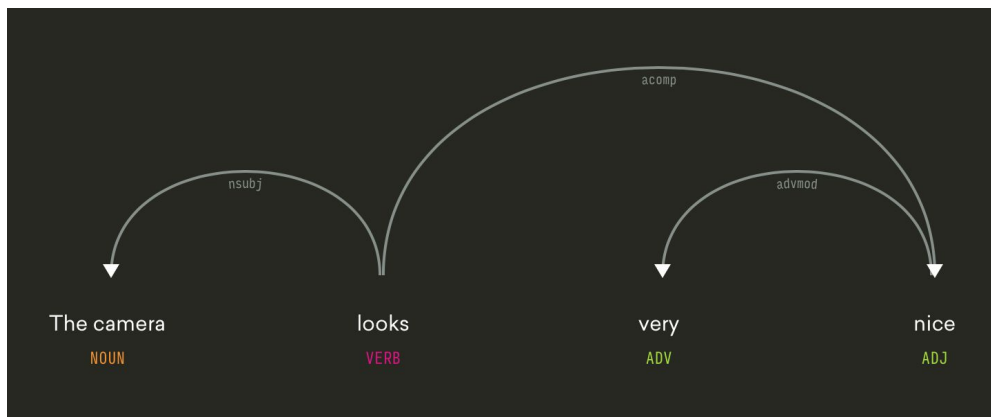


Figure 10. An example of Rule 1.

Rule 1 means that given a sentence where a noun or noun phrase is connected by a "nsubj" relation to a verb, and the same verb is connected by an "acomp" relation to an adjective, with possible negation or modifiers included, the noun/noun phrase, modifiers, and adjective will be extracted as a feature-modifier-opinion triplet respectively.

Rule 2: $\text{nsubj}_{\text{verb} \rightarrow \text{f}} + (\text{neg}) + (\text{advmod}) + \text{advmod}_{\text{f} \rightarrow \text{o}}$

One example of this pattern is:

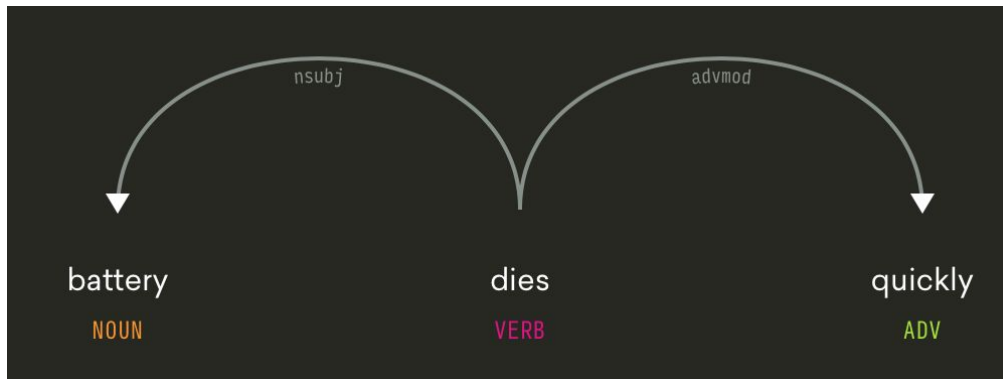


Figure 11. An example of Rule 2.

Rule 2 means that given a sentence where a noun or noun phrase is connected by a "nsubj" relation to a verb, and the same verb is connected by an "advmod" relation to an adverb, with possible negation or modifiers included, the noun/noun phrase, modifiers+adverb, and verb will be extracted as a feature-modifier-opinion triplet respectively.

Rul3 3: (neg) + (advmod) + $xcomp_{o \rightarrow f}$

One example of this pattern is:

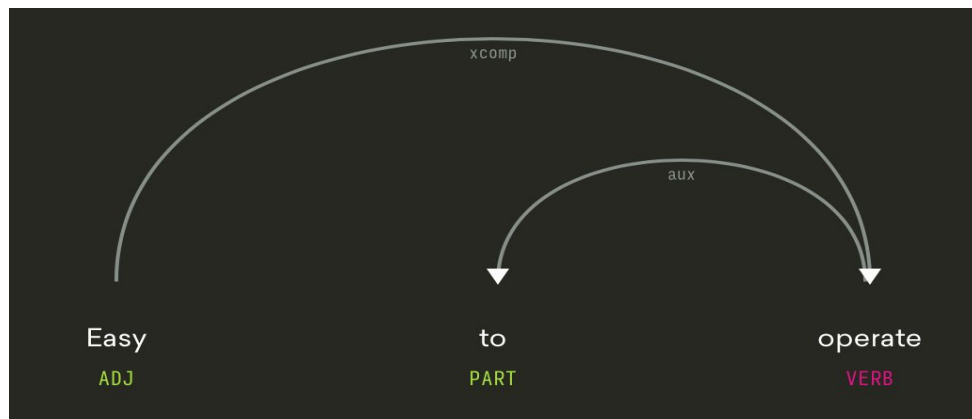


Figure 12. An example of Rule 3.

Rule 3 means that given a sentence where a verb is connected by a "xcomp" relation to an adjective, with possible negation or modifiers included, the verb, modifiers, and adjective will be extracted as a feature-modifier-opinion triplet respectively.

In this thesis, features could be noun, noun phrase or verb. Nouns and noun phrases are located with the help of Spacy "doc.noun_chunk" function. The "noun_chunk" here refers to a noun with several words describe the noun. The verb features are located by POS tagging.

Based on the rules above, the feature-opinion connections can be extracted and then added to the candidate keyphrase set.

3.2.2.3 Pruning

Due to the variety of English text expressions, it is possible to extract some wrong feature-opinion connections by dependency relation analysis. Therefore, a pruning process has to be adopted.

First, the pruning process calculates the occurrence of each feature-opinion pair and then sorts them by frequencies. Generally, the higher the frequency of a candidate keyphrase, the less likely it is to be a wrong one. This thesis sets a threshold for candidate phrases and discards the candidates whose frequency are less than the threshold. The rest of the candidates have a higher chance of becoming keyphrases of the reviews. However, there are a lot of semantically similar near-duplicate phrases in the candidate set, so a clustering process is needed to obtain the final semantically different keyphrases.

3.3 LDA-Based Keyphrase Clustering and Ranking.

As described in Section 3.2, the quality of feature and opinion extraction will directly affect the final mining results. In general, the statistical characteristics of words are usually taken into account when extracting the feature words and the opinion words. The statistical features of words include the word term frequency (TF), the inverse document frequency (IDF), the first occurrence position of word, and even the length of the word. This kind of information is easy to acquire, but it often has some limitations. To obtain more accurate extraction results, semantic information of words should be taken into account.

The semantic information of words includes attributes that describe the meaning of the word, such as the part of speech and synonyms. Some semantic information can be acquired from external resources, such as WordNet [30], Wikipedia [50], synonym dictionary [42] and search engines [19]. Such kinds of methods use synonyms to express the semantic similarity between words. On the other hand, semantic information can also be acquired within the document, such as dependency relations, part of speech and latent semantics.

Topic models aim to find the latent semantic information from the text. In recent years, topic modeling has been widely used in various tasks related to text analytics and information retrieval, such as topic extraction, document clustering, and text classification. Topics refer to central ideas that are expressed in a document, which are mainly composed of some related feature words. The feature words here do not refer to product features, but to a group of words that usually appear together. For example, if an article has a topic “education”, words such as “teacher”, “textbook”, “student”, “scholarship” may often appear, while the words “car” or “Christmas” are

unlikely to appear. However, in topic models, words can belong to multiple groups with different levels of membership, for example because some words are related to multiple different central ideas.

The topics found by a topic model are dependent on the corpus, which means given a different corpus, the hidden topics are different. Also, topics are a highly abstract and compressed representation of semantics in the corpus. As a formal mathematical definition, a topic is described as a conditional probability distribution over words related to the topic. The more closely a word is related to a particular latent semantic topic across a corpus of articles, the higher the probability will be for that word in the conditional distribution of words in that topic.

Due to the good mathematical foundation and flexible range of expansions of topic models, they were immediately paid attention by many scholars and were widely used in various text mining and information processing tasks. So far, a Google scholar search has shown that the number of references to the topic model LDA has already exceeded 31000. In addition, the research on LDA-based variant models is becoming increasingly popular, contributing to the development of topic models.

The basic concept of the topic models is to locate the topics of the corpus by parameter estimation, and the prevalence of topics in each document and prevalence of words in each topic are all represented by multinomial probability distributions which can be used to dig the deep semantic information from the text. Currently, some popular models include LSI, PLSI, and LDA. However, this thesis chooses the LDA model, which is more flexible than other topic models such as PLSA and LSI.

In this thesis, candidate keyphrases are first extracted according to the dependency relation rules. The candidate keyphrases are in the form of <feature, modifier, opinion>. Then, topic modeling was applied on the candidate keyphrases to cluster the keyphrases. This thesis regards one keyphrase as a document, and each document can have multiple topics. However, we only select the largest possible topic to cluster the documents. Under the concept of topic modeling, each word is generated by a specific topic-word distribution.

It is easy to understand that the key information of a document is closely related to the topics of the document, so the final results should be able to represent the topics of the documents.

3.3.1 LSI and PLSA

LSI (Latent Semantic Indexing) is a method to define the latent semantic relations by using the co-occurrence information between words in a document. LSI was proposed mainly to solve the problem of semantic mismatch in information retrievals, such as

ambiguity and polysemy. LSI is not strictly a topic model because it is not a probabilistic generative model. LSI mainly utilize the SVD (Singular Value Decomposition) to decompose the document-word matrix to to a product of low rank matrices, which can significantly reduce the dimensions of document representation.

In LSI model, all the documents in the corpus are represented by a document-term matrix A with M rows and N columns, M means the number of entries in the dictionary, and N indicates the number of documents in the corpus. Each row of matrix A represents one word, and each column represents one document. An example is shown in Figure 13.

$$A = \begin{pmatrix} a_{11} & \dots & a_{1j} & a_{1N} \\ \dots & & & \dots \\ a_{i1} & & a_{ij} & a_{iN} \\ \dots & & & \dots \\ a_{M1} & \dots & a_{Mj} & a_{MN} \end{pmatrix}$$

Figure 13. LSI document-term matrix

Here a_{ij} is the weighted word frequency (e.g., TF/IDF) of word i that appears in document j . Obviously, for a large corpus, the dimensions of matrix A will become very high. However, by using SVD matrix, A can be decomposed into three low dimensional matrices, which can be represented as $A=TS^{\frac{1}{2}}D^{\frac{1}{2}}$. T is a $M \times K$ word vector matrix, which represents the association (or relevance) weights of each word to each topic. S is a $K \times K$ diagonal matrix, in which K can be understood as the number of topics. D is a $K \times N$ document vector matrix that represents the association (or relevance) weights of each topic to each document in the corpus. Figure 14 illustrates the decomposition.

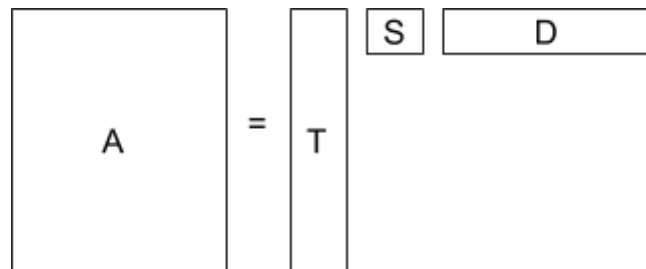


Figure 14. SVD decomposition process

Through the LSI model, a set of mathematically orthogonal topics can be found. A document can be represented as a set of weights over different topics. Within each topic, the vocabulary can also be sorted according to their relevance weight in the topic, thus visualizing the latent semantic information of the document.

However, one weakness of LSI is that the computation cost of SVD is very high, and also when new documents are coming, the model needs to be updated and re-trained, which can consume a lot of system resources.

On top of LSI model, Hofmann et al. [55] proposed a probabilistic generative model, called PLSA. The difference between LSI and PLSA is that PLSA introduces the concept of probability, which simulates the generation of feature words in the corpus. Furthermore, PLSA model defines the concept of “hidden topic”. “Topic” is a latent trend of words in the generative process.

Using a graphical plate model [76] to describe PLSA as in Figure 15 is convenient. A probabilistic graphical model is diagram that represents statistical dependency relationships between variables as a directed graph. In Figure 15, the shadow nodes represent observable variables, white node represents latent variables. A box represents that its internal structure will be repeatedly sampled, and the subscript in the lower right corner indicates the number of repetitions. The arrows between variables indicate the probability dependency.

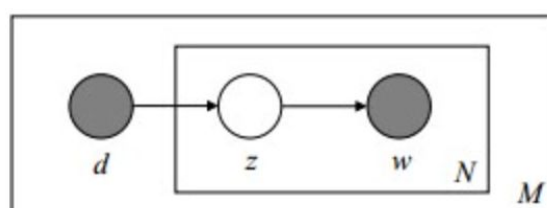


Figure 15. PLSA model

Here d represents a document, z denotes a topic index of a particular word to be generated into the document, w denotes the vocabulary index of the generated word, and N is the number of words in document d . M represents the number of documents in the corpus. The PLSA model contains two kinds of conditional probability distributions, namely, the document-topic distribution $p(z|d)$ and the topic-word distribution $p(w|z)$. Through these two distributions, PLSA can generate all the documents in the corpus.

The core problem of PLSA model is to solve two distributions $p(z|d)$ and $p(w|z)$. In Hofmann’s paper, the expectation maximization (EM) algorithm was used to find values for these unknown distributions that best fit the observed document content according to a maximum likelihood criterion.

However, although PLSA is a probabilistic generative model for the set of training documents, it is not a complete generative model for new documents not seen in the training phase. This is because $p(z|d)$ are directly estimated from the data as model

parameters separately for each training documents, and no general model is learned about $p(z|d)$ over the documents.

PLSA uses the training set to fit the $p(z|d)$ distributions, whereas for documents outside the training set, these distribution cannot be directly applied. Moreover, although PLSA fits a common set of $p(w|z)$ distributions to all documents, it fits a separate $p(z|d)$ distribution with its own parameters to each training document, which may lead to an overfitting problem. In other words, PLSA model does not good at predicting new documents.

3.3.2 LDA

On top of PLSA, Blei [56] proposed Latent Dirichlet Allocation (LDA) in 2003, which is a three layer “document-topic-word” Bayesian generative model. The generation process of LDA is similar to PLSA model, and the only difference is that LDA model considers the parameter θ (document-topic distribution) and ϕ (topic-word distribution) also as variables, adding a Dirichlet prior to each of them. The choice of Dirichlet distribution as a priori distribution of θ and ϕ is because Dirichlet distributions and multinomial distributions are conjugate so that it can be convenient for model computation.

LDA is an unsupervised machine learning algorithm that can be used to identify topic information in a large document corpus. Like LSI and PLSA, the LDA model is based on a model of a document as a bag of words, ignoring the order of any sentences or words. That is, LSI, PLSA, and LDA all use a bag of words model to convert each document to a word frequency vector so that the text information can be transformed into counts of words, which is easier for a computer to analyze.

Figure 16 shows the graphical plate model representation of LDA. Here w represents a word in the document, z is the topic assignment for w . θ is the topic distribution of the document. α is the Dirichlet-prior parameter of the per-document topic distribution. ϕ is the word distribution for topic z . β is the Dirichlet-prior parameter of the per-topic word distribution. N is the number of words in the document, while K is the total number of topics and M is the number of documents in the corpus.

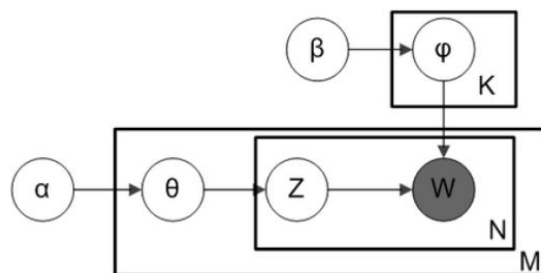


Figure 16. LDA model

The generative process of LDA model is as following:

Suppose there are two bottles of dice. The first bottle represents the generation of topic distributions for documents and it has a lot of document-topic dice inside. The second bottle represents the generation of words from topics and it has a lot of topic-word dice inside. A document-topic die has K sides, every side represents one topic. A topic-word die has V sides, every side represents one word in the corpus dictionary. The dice are loaded so that the probabilities of different sides are nonuniform and are different in each die.

First, randomly select K topic-word dice from the second bottle, and then:

1. For each document in the corpus:

Randomly select a document-topic die from the first bottle

2. For each word in the current document:

First, throw the document-topic die to get a topic index z ; Then, choose the z :th topic-word die from the K topic-word dice and throw it to get a word index w .

The difference between PLSA and LDA is that, in PLSA, the document-to-topic distributions and the topic-to-word distributions are fitted to specific values for training data. However, in LDA, the topic-to-word distribution and document-to-topic distribution are samples from the respective two Dirichlet-prior distributions, and new documents can get new samples of document-to-topic distributions, and moreover a full posterior distribution can be computed for the parameters. In other words, LDA is a Bayesian version of PLSA model.

Similarly, the core problem of LDA model is to find well-fitting document-topic distributions θ and topic-word distributions ϕ , or more generally their posterior distributions. Currently, there are two kinds of methods to infer θ and ϕ , which are the variational-based method and sampling-based method [77]. The sampling algorithm constructs an empirical distribution to approximate the posterior distribution after collecting the document samples. The most popular sampling algorithm for topic modeling is Gibbs sampling [78].

In contrast to the sampling-based method that approximates the target distribution (θ and ϕ) based on document samples, the variational-based method proposes a simplified functional form of a probability distribution to approximate the real posterior distribution and then optimizes the parameters of the functional form to represent most of the probability mass of the posterior distribution. The problem is then transformed into an optimization problem. In Blei's paper, the combined method of variational inference and EM algorithm are used to infer the parameters of LDA

model. Similar algorithms include Hoffman's Online variational inference [79]. In this thesis, the "ldamodel" module in the Gensim¹⁸ Library is used as a tool to solve the LDA model, which implements Hoffman's algorithm in Python.

3.3.3 Clustering Keyphrases Based on LDA Model

As previously explained, LDA can be used as a clustering algorithm for words, while the topics can be considered as product features. This thesis uses the LDA model, regarding each keyphrase as a document and then applying LDA to model the text.

The keyphrases in this thesis are in the form of <feature, modifier, opinion>. Thus, it can be seen that each keyphrase contains one and only one product feature. Therefore, by applying LDA model, each keyphrase will be mapped into different product feature categories. The entire clustering process is shown in Figure 17.

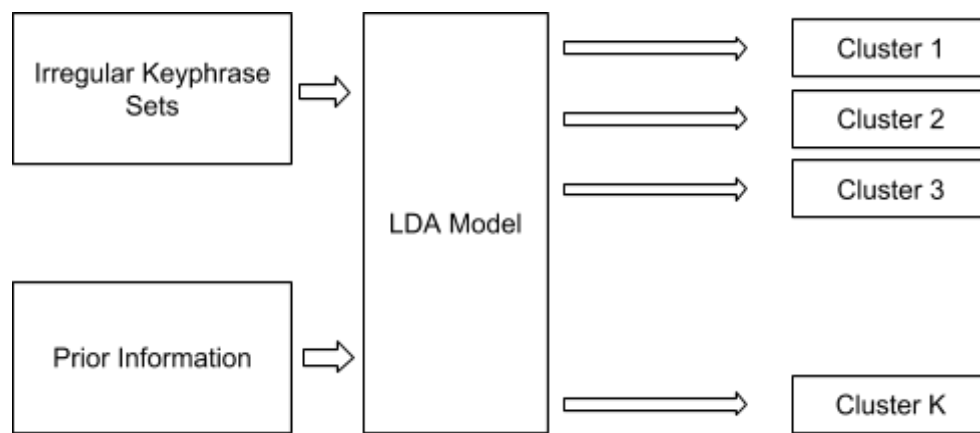


Figure 17. Clustering process

There are several problems need to be solved before the clustering process. First, the determination of K . K is the number of topics as well as the number of clusters. In this thesis, K is determined by experiment as will be described in Chapter 4.

The second problem is the determination of Dirichlet-prior parameters α and β . More specifically, α affects the extent to which topics differ between documents. A larger α value usually means every document in the corpus contains most of the topics from the topic set, which will lead to documents being more similar regarding what topics they contain. β affects the extent to which words differ between topics. Similarly, a larger β value usually means every topic contains most of the words in the dictionary, which will lead to topics being more similar regarding what words they contain. A heuristic method found that a good model usually has $\alpha=50/K$, $\beta=0.01$, which is proposed by Gregor Heinrich [80] in his paper *Parameter estimation for text analysis*.

¹⁸ <https://radimrehurek.com/gensim/>

After setting the above mentioned hyperparameters of the LDA model, the keyphrases are ready to be clustered. Regarding all the keyphrases as the corpus, a corpus dictionary can be constructed. Every keyphrase is regarded as a document and is transformed into a vector. Passing the vectors into LDA model as input and train the model. Finally, LDA will return two important distributions, namely θ and ϕ , which are document topic multinomial distribution and topic word multinomial distribution respectively.

θ is a $M \times K$ matrix, and each row corresponds to a document from the corpus, θ_{mk} represents the possibility of a topic k in document m . ϕ is a $K \times N$ matrix in which each row corresponds to a topic in the topic set. ϕ_{kn} represents the probability of word n in topic k . Figure 18 illustrates the notation.

$$\theta = \begin{bmatrix} \theta_{11} & \theta_{12} & \dots & \theta_{1K} \\ \theta_{21} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \theta_{M1} & \dots & \dots & \theta_{MK} \end{bmatrix}, \quad \phi = \begin{bmatrix} \phi_{11} & \phi_{12} & \dots & \phi_{1N} \\ \phi_{21} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \phi_{K1} & \dots & \dots & \phi_{KN} \end{bmatrix}$$

Figure 18. Matrix presentation of θ and ϕ

The detailed flow chart of the LDA based clustering process is shown in Figure 19. After getting θ , keyphrases can be clustered. Each document is a row in θ which contains the probabilities of different topics in the document. Then, select the topic with the highest probability as the representative topic of the current document, and classify all the keyphrases according to their representative topics.

More specifically, for a document m , the document-to-topic multinomial distribution assigned to m is θ_m , for example which may equal to $[0.8, 0.05, 0.1, 0, 0, 0.05]$. The total number of elements in the list is K , and in the example case $K=6$. All the elements should sum up to 1. In this case, the first topic has the highest probability, so the document will be clustered into cluster 1.

However, when K has a large value, it is possible that some topic clusters do not have any keyphrases in them. This is because not all the topics have the chance to be the representative topic of the keyphrases.

In this thesis, each triplet $\langle \text{feature}, \text{modifier}, \text{opinion} \rangle$ is regarded as a document, and each word in the document is treated separately as vocabulary-word. For example, $\langle \text{"picture quality"}, \text{"not"}, \text{"good"} \rangle$ contains 4 occurrences of vocabulary-words, which are "picture", "quality", "not", "good".

As mentioned in the previous section, topics can be considered as product features, so the final results are the keyphrases clustered based on product features.

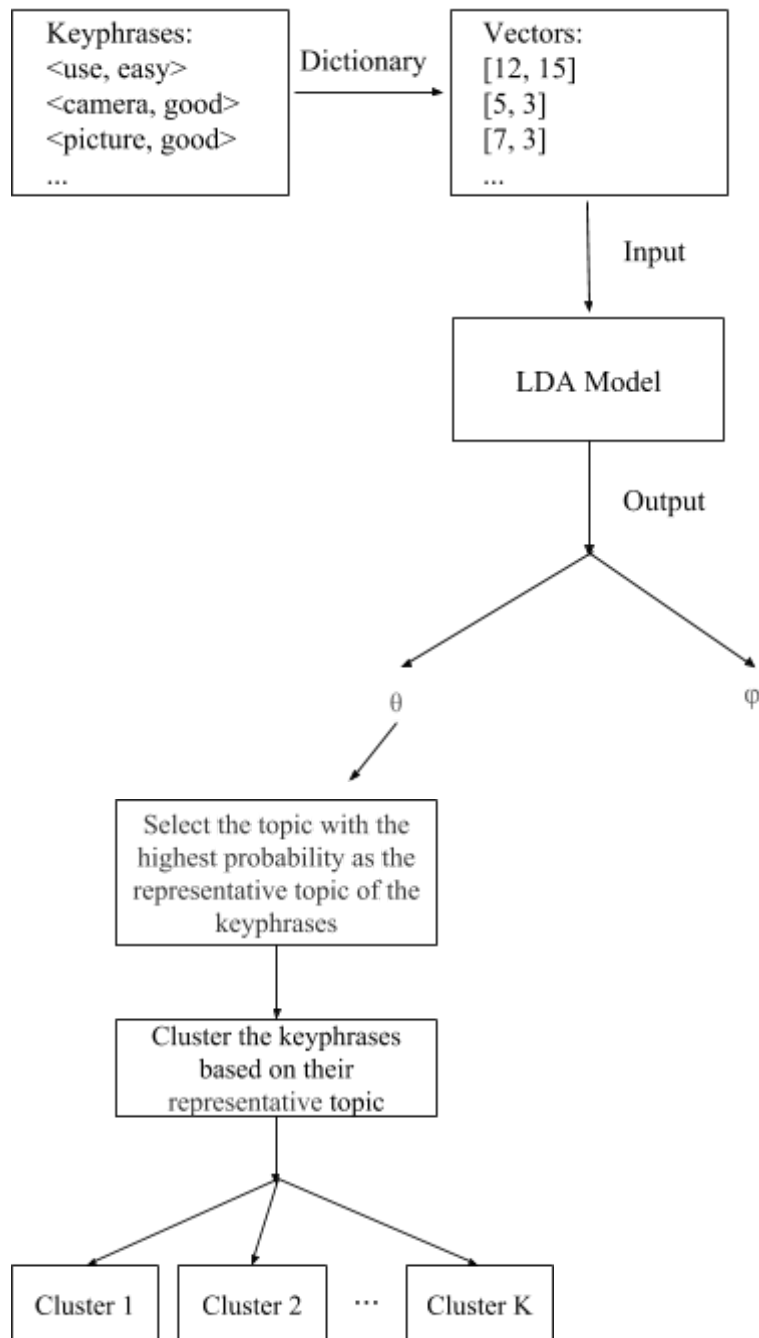


Figure 19. LDA clustering process flow chart

3.3.4 Keyphrase Ranking

There could be tens of hundreds of keyphrases in one cluster, and if customers want to have an integral understanding of each cluster, they still need to browse all the keyphrases in each cluster, which will cost a lot of time. In addition, it is difficult to

display all the keyphrases, especially on a small screen platform like mobile phone. One way to solve this problem is to summarize the clusters by automatically generating a semantic tag for each keyphrase cluster.

In this thesis, a sorting process is applied for the summarization of clusters. By selecting the most representative keyphrases as the tags of clusters, the efficiency of user obtaining product information is effectively improved.

The most important goal of this chapter is to select the most representative keyphrases to be the tag of the clusters, and the selected representative keyphrases should cover well the content of the corresponding clusters. The problem is formalized as follows: By keyphrase clustering process a set of clusters is obtained, which is $V = \{v_1, v_2, v_3, v_4, \dots, v_K\}$. Each cluster v contains a set of keyphrases $v = \{p_1, p_2, p_3, \dots, p_n\}$, n is the number of keyphrases in the cluster. Each keyphrase can be regarded as a set of words. The goal of this chapter is to extract one best representative keyphrase p_t to be the tag of cluster v , and $p_t \in v$.

How to sort the keyphrases by their representativeness or importance is a crucial problem to be solved in this chapter. This thesis presents two different methods to measure the importance of keyphrases, and then applies them in the ranking process. The methods include LDA-TFIDF and LDA-MT, which will be explained in the next section.

3.3.4.1 LDA-TFIDF

LDA-TFIDF is an algorithm that combines the LDA model with TF-IDF score. TF-IDF is a popular method for evaluating the importance of a word in a document. TF is the Term Frequency, which is based on the number of times a word appears in a document. IDF is the Inverse Document Frequency, which is based on the number of a word appears in different documents. TF-IDF is a statistical method, and its basic concept is that the importance of a word to a document is proportional to the frequency that it appears in the current document, and is the inverse proportion to the frequency that it appears in all documents. Taking camera reviews as an example, words like “it”, “the” have a high frequency in all review sentences, so their ability to distinguish topics is feeble. On the contrary, words like “Screen”, “photo” only appear in some sentences, which means these words have a higher probability to be topic words.

In the following t_i denotes a term, and d_j represents a document. The term frequency $TF(t_i, d_j)$ shows the frequency of t_i appearing in d_j , document frequency $DF(t_i)$ represents the logarithm of the frequency that t_i appears in all documents. IDF is the reciprocal of DF . TF-IDF is the product of term frequency and inverse document frequency. The calculation formulas are shown below, where n_{t_i, d_j} represents the

number of times that t_i appears in document d_j , $|D|$ represents the total number of documents in the corpus, and $|\{j: t_i \in d_j\}|$ represents the number of documents that contain t_i .

$$TF(t_i, d_j) = \frac{n_{t_i, d_j}}{\sum_{t_k \in d_j} n_{t_k, d_j}} \quad (3-1)$$

$$DF(t_i) = \log \frac{|\{j: t_i \in d_j\}|}{|D|} \quad (3-2)$$

$$IDF(t_i) = \frac{1}{DF(t_i)} \quad (3-3)$$

$$TF-IDF(t_i, d_j) = TF(t_i, d_j) * IDF(t_i) \quad (3-4)$$

Since different clusters should describe different topics, and keyphrases in the same cluster should have the similar topics, this thesis takes each cluster v as a document and regards all the clusters V as a corpus. For each keyphrase, first calculate the TF-IDF score for each word it contains, taking into account that longer keyphrase should have more information, the TF-IDF score of the whole keyphrase is the sum of the TF-IDF score of the words it contains.

Using p to represent keyphrases, w_i denotes the words in p , s_{w_i} represents the TF-IDF score of w_i , and the TF-IDF score of p , which is s_p , are calculated as:

$$s_p = \sum_{w_i \in p} s_{w_i} \quad (3-5)$$

Where

$$s_{w_i} = TF-IDF(w_i, v) \quad (3-6)$$

The representative tag of v , which is denoted as p_t , is selected according to the following equation:

$$p_t = \operatorname{argmax}_{p \in v} s_p \quad (3-7)$$

For each cluster, select the representative keyphrase based on the above formula, and finally get K-representative keyphrases.

3.3.4.2 LDA Max Topic (LDA-MT)

Unlike the LDA-TFIDF method, the LDA-MT method does not use TF-IDF score as the criterion, but rather a topic relevance score as the standard to rank the keyphrase. The topic relevance score is calculated according to another important output from LDA model, which is the topic-word distribution ϕ . ϕ is a $K \times N$ matrix that describes the probability of each word to occur in each topic. For example, ϕ_{kn} represents the probability for the n th word in the dictionary to occur in topic k .

The basic concept of LDA-MT is that for a keyphrase p in cluster v , if the topic of each word contained in p is more relevant to the topic of the whole cluster v , then p should have stronger ability to describe v . In this situation, p should have a higher topic relevance score. Also, considering that longer keyphrase should contain more information, the final score of p should be the sum of the words' topic relevance score.

The formula for calculating the topic relevance score of p is as follows, in which z represents the topic of cluster v .

$$s_p = \sum_{w_i \in p} s_{w_i} \quad (3-8)$$

Where

$$s_{w_i} = \phi_{zw_i} \quad (3-9)$$

Similarly, the representative tag p_t is selected according to the following formula:

$$p_t = \underset{p \in v}{\operatorname{argmax}} s_p \quad (3-10)$$

4. Evaluation

This chapter aims at evaluating the keyphrase extraction system which is described in the previous chapter, and compares the results obtained by two different algorithms LDA-TFIDF and LDA-MT. To evaluate the automatic extraction result, a manual extraction process is applied in advance as will be described in Section 4.3. The evaluation results prove the feasibility and effectiveness of the automatic extraction system proposed in this thesis.

4.1 Data Set

The experiment data are reviews of two camera products Kodak PIXPRO AZ251 and Sony Cyber-Shot DSC-RX100 crawled from the Amazon website, and each product has 70-100 reviews. All reviews are stored in the local database.

4.2 Evaluation Criteria

The evaluation of the keyphrase extraction system is an evaluation of a natural language understanding system. However, the problem is that natural language is difficult to transform into structured information. Moreover, understanding of natural language is a subjective reflection made by the human brain, which means that different people can have a different understanding of the same text. Therefore, it is difficult to create objective criteria to evaluate a natural language understanding system.

However, this thesis employs Precision to evaluate the proposed keyphrase extraction system, which is commonly used in natural language processing. The calculation formula of Precision is:

$$Precision = \frac{\text{Number of correctly extracted keyphrases}}{\text{Number of extracted keyphrases}} \quad (4-1)$$

In the above equation, the number of correctly extracted keyphrases is computed by comparing extracted keyphrases to the manually extracted keyphrases, in a similar evaluation system as done by Turney [81]. The detailed criteria are to go over each extracted keyphrase and score it as follows: 1) if the keyphrase extracted by the system exactly matches one of the manually extracted keyphrases in words, then the number of correct keyphrases is increased by 1; 2) If the keyphrase extracted by the system best matches some keyphrase in the manual result so that they do not exactly match in words, but they have very close meaning as manually judged by the author of this thesis, for example, “great picture quality” and “perfect image”, then the number of correct keyphrases is increased by 0.7; 3) Otherwise the number of correctly extracted keyphrases is not increased.

4.3 Result Analysis

Kodak PIXPRO AZ251 has 78 reviews in total, and Sony Cyber-Shot DSC-RX100 has 97 reviews in total. First, manual extraction will be applied to both datasets. The basic manual extraction principle is to extract only explicit features with their opinion words, and the keyphrases that are semantically similar will be merged. Next, all reviews are passed into the review extraction system, and the extraction results are compared to the manual results as described in the previous section; the closer the results are to the manual results according to the precision measure, the more accurate the extraction system is considered to be.

For Kodak PIXPRO AZ251, the relationship between system precision and the number of extracted topics K is shown in Figure 20:

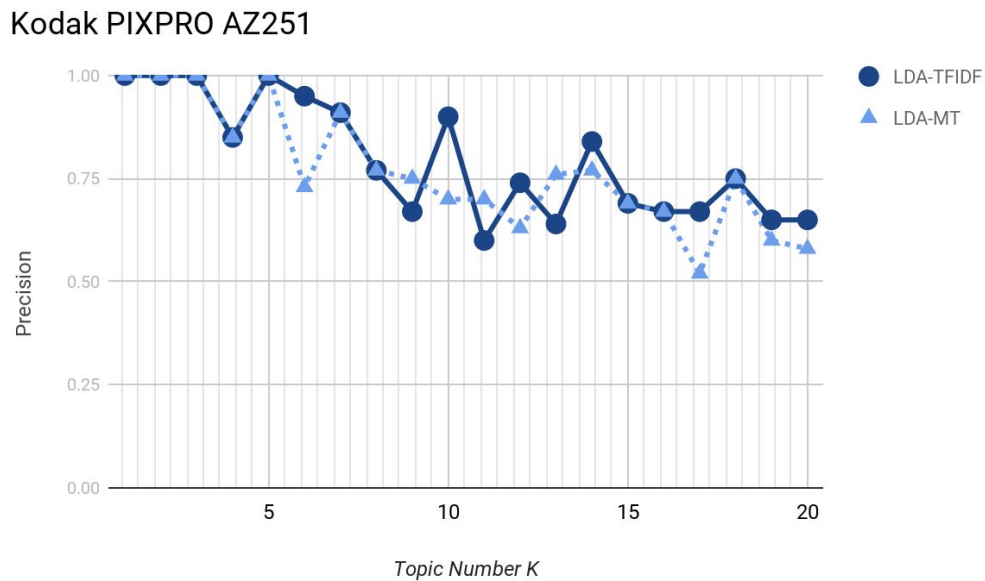


Figure 20. Result of Kodak PIXPRO AZ251

For Sony Cyber-Shot DSC-RX100, the relationship between system precision and the topic number k is shown in Figure 21:

Sony Cyber-Shot DSC-RX100

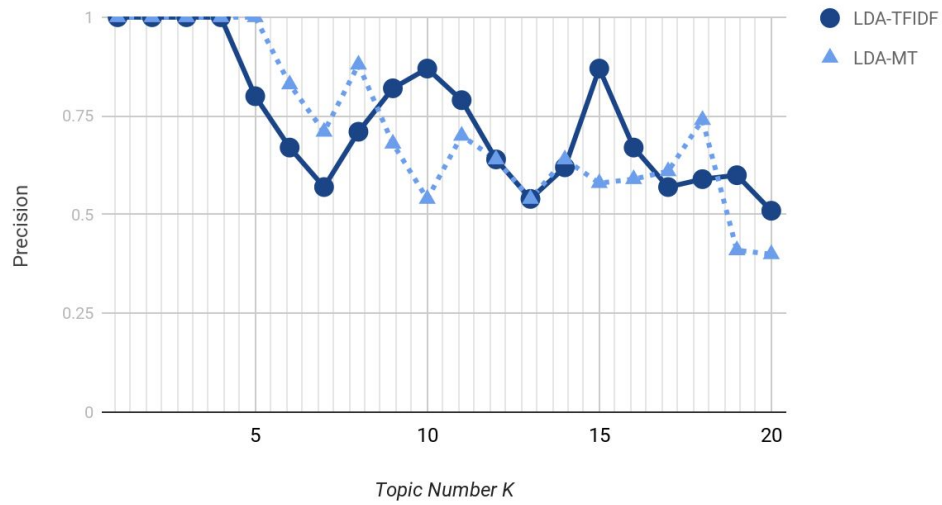


Figure 21. Result of Sony Cyber-Shot DSC-RX100

By observing the above two line charts, it can be seen that when K is smaller, the system has higher precision, and with the increase of K , the system precision is gradually reduced. Such trend is understandable because a large K indicates that the system will extract more keyphrases, then the probability of erroneously extracted keyphrases will also increase. However, in practice, it is pointless to let K take either a too small or a too large value. When K is too small, the system can only extract a few keyphrases, so that customers can not get a comprehensive understanding of the product; When K is too large, the precision of the system is greatly reduced, and a large number of faulty keyphrases will also interfere with the customers.

For Kodak PIXPRO AZ251, it is found that when $K=10$, LDA-TFIDF algorithm obtains higher precision, and LDA-MT algorithm obtains a higher precision when $K=7$. Overall, the average precision of LDA-TFIDF algorithm is better than LDA-MT algorithm although there is a lot of variability and the difference may not be statistically significant.

For Sony Cyber-Shot DSC-RX100, it is found that when $K=15$, the LDA-TFIDF algorithm obtains higher precision, while LDA-MT algorithm obtains higher precision in $K=8$. Overall, LDA-MT algorithm is superior to LDA-TFIDF algorithm when K is between 5-10, and when K is between 10-20, both algorithms have advantages and disadvantages; these results again show much variability and more tests would be needed to establish statistical significance of the differences.

Combining the results of two camera products, it is found that the average precision of LDA-MT algorithm seems to be inferior to the LDA-TFIDF algorithm for larger

values of K . One possible reason is that when K has larger value, the system will have more clusters. The LDA-MT algorithm chooses the overall highest-probability topic as the topic of the document, and then the keyphrases are sorted according to the sum of the contribution of each word to that corresponding topic. When there are many topics, each word has a higher probability of being assigned to a wrong topic if the overall highest-probability topic of the keyphrase is not suitable to that individual word, which resulting in a lower overall accuracy. Instead of being based on topic contribution, the LDA-TFIDF algorithm is based on the TFIDF score of the words, thus with higher K value it still works well.

Table 2 shows some example keyphrases extracted from the two camera products by the proposed system.

Product Name	LDA-TFIDF	LDA-MT
Kodak PIXPRO AZ251	Easy to use Flash work wonderfully Disappoint purchase Picture quality excellent Seller horrible Lens cap loose Low light level photo terrible	Seller unhelpful Flash work wonderfully Low light level photo terrible Battery die quickly Right price
Sony Cyber-Shot DSC-RX100	Battery drain quick Menu not intuitive Image quality good 4k video sharp User interface not user-friendly 20 fp rate amaze Quality not acceptable	Camera size perfect User interface not user-friendly 4k video sharp Battery life disappoint Image quality good Small menu option complicate

Table 2. Extraction result examples

5. Implementation of the Keyphrase Extraction System

On top of the knowledge from the previous chapters, an automatic keyword extraction system is designed and implemented in this thesis. This system includes a crawler module, an extraction module and a web interface module. The system architecture is shown in Figure 22.

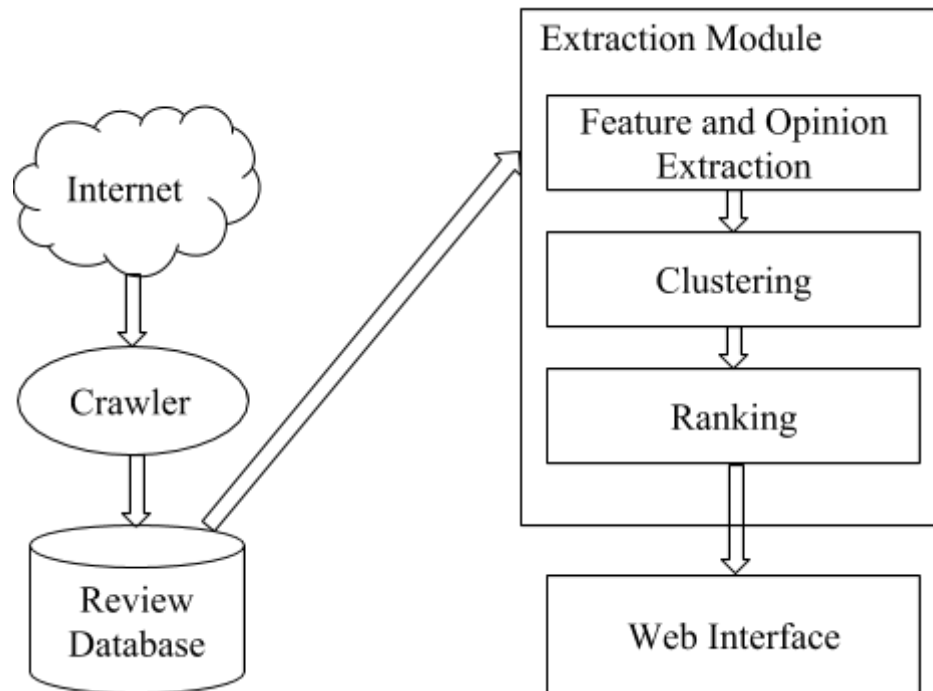


Figure 22. System architecture

The crawler module, as well as the extraction module, are written in Python. The web interface module is implemented with HTML, javascript, and node.js.

5.1 Crawler Module

The main task of crawler module is to crawl Amazon reviews. The module is implemented based on Scrapy¹⁹, a Python crawler library. By analyzing the DOM structure of product review page, the review related information can be extracted. Figure 23 shows the DOM structure of a typical review page, where red boxes are the contents that should be crawled.

¹⁹ <https://scrapy.org/>

```

::before
<span data-hook="total-review-count" class="a-size-medium totalReviewCount">677</span>
::after
</div>
</div>
::after
</div>
</div>
::after
</div>
<div class="a-row averageStarRatingNumerical">
  ::before
  <span class="a-declarative" data-action="a-popover" data-a-popover="{"inlineContent":"Amazon calculates a product's star ratings using a machine learned model instead of a raw data average. The machine learned model takes into account factors including: the age of a review, helpfulness votes by customers and whether the reviews are from verified purchases."}">
    <a href="javascript:void(0)" class="a-popover-trigger a-declarative">
      <span data-hook="rating-out-of-text" class="a-pop-rating-out-of-text">4.6 out of 5 stars</span>
      <i class="a-icon a-icon-popover"></i>
    </a>
  </span>
  ::after
  </div>
  <div class="a-row a-spacing-top-small"></div>
</div>
<div class="a-fixed-left-grid-col a-col-right" style="padding-left:2%;float:left;">
  <div class="a-fixed-left-grid">
    <div class="a-fixed-left-grid-inner" style="padding-left:55px">
      ::before
      <div class="a-text-center a-spacing-top-micro a-fixed-left-grid-col product-image a-col-left" style="width:65px;margin-left:-65px;float:left;"></div>
      <div class="a-fixed-left-grid-col product-info a-col-right" style="padding-left:2%;float:left;">
        <div class="a-row product-title">
          ::before
          <h1 class="a-size-large a-text-ellipsis">
            <a data-hook="product-link" class="a-link-normal" href="/Nikon-D3300-AF-P-18-55mm-Digital/001JN2800A/ref=cm_cr_arp_d_product_top?ie=UTF8">Nikon D3300 w/ AF-P DX 18-55mm VR Digital SLR - Black</a>
          </h1>
          </div>
          <div class="a-link-normal" title="5.0 out of 5 stars" href="/gp/customer-reviews/R1908BCG1HE55U/ref=cm_cr_arp_d_rvw_ttl?ie=UTF8&asin=001JN2800A"></div>
          <span class="a-letter-space"></span>
          <a data-hook="review-title" class="a-size-base a-link-normal review-title a-color-base a-text-bold" href="/gp/customer-reviews/R1908BCG1HE55U/ref=cm_cr_arp_d_rvw_ttl?ie=UTF8&asin=001JN2800A">Excellent for beginners. A few suggestions on accessories.</a>
          ::after
          </div>
        </div>
        <div class="a-row">
          ::before
          <span data-hook="review-author" class="a-size-base a-color-secondary review-byline">
            <span class="a-color-secondary">By</span>
            <span class="a-letter-space"></span>
            <a data-hook="review-author" class="a-size-base a-link-normal author" href="/gp/profile/amzn1.account.AF5R5ZKUT5ZU2WBSLKSG6DC2MDAQ/ref=cm_cr_arp_d_pdp?ie=UTF8">Adam P.</a>
          </span>
          <span class="a-letter-space"></span>
          <a class="a-link-normal enthusiast-badge aok-nowrap" href="/gp/profile/amzn1.account.AF5R5ZKUT5ZU2WBSLKSG6DC2MDAQ/ref=cm_cr_arp_d_pdp_enh?ie=UTF8">Top Contributor: Pets</a>
          <span class="a-declarative" data-action="cr-popup" data-cr-popup="{"width":"340","title":"Help","url":"/gp/help/customer/display.html/ref=cm_cr_dp_bdg_help?ie=UTF8&nodeId=14279681&pop-up=1#tr","height":"340"}"></span>
          <span class="a-letter-space"></span>
          <span data-hook="review-date" class="a-size-base a-color-secondary review-date">on February 28, 2017</span>
          ::after
          </div>
        </div>
        <div class="a-row a-spacing-mini review-data review-format-strip"></div>
        <div class="a-row review-data">
          ::before
          <span data-hook="review-body" class="a-size-base review-text">
            "Update: For those looking to capture action photography when the subject is not near the camera (e.g., throwing a toy as a dog is running away from you, wildlife and birds), I highly suggest looking into the following lens:<br>
            <a data-hook="product-link-linked" class="a-link-normal" href="/Nikon-AF-S-DX-NIKKOR-55-300mm-f-4-5-5-6G-ED-Vibration-Reduction-Zoom-Lens-us-for-Nikon-DSLR-Cameras/0003ZSHNCC/ref=cm_cr_arp_d_rvw_txt?ie=UTF8"></a>
            "
            <br>
            <br>
            "Great camera, I purchased this mainly to take action shots of my dog. Luckily I purchased this around Black Friday and it was considerably cheaper than the normal selling price. I love this camera, I have no issues using it and by exporting the images as RAW format I am able to do post-processing in Adobe Softwares (i.e. Photoshop, Lightroom)."
            <br>
            <br>
          </span>
        </div>
      </div>
    </div>
  </div>

```

Figure 23. The DOM structure of Amazon review page

The local database has two collections, one is the product collection, the other is the review collection. The structure of the product collection is shown in Table 3, and the structure of the review collection is shown in Table 4.

Key	Type	Description
_id	Object	Product unique id

asin	String	Amazon product unique id
title	String	Product title
url	String	Product page url
rate	Double	The overall rating of the product
scrap_date	Date	The date that the product is crawled
num_of_ops	Int32	The total number of reviews of the product

Table 3. The dictionary of the product collection

Key	Type	Description
_id	Object	Review unique id
author	String	Review author
title	String	Review title
item	Object	Product id
rate	Double	Review rating
date	Date	Review date
opinion	String	Review content

Table 4. The dictionary of the review collection

5.2 Extraction Module

The extraction module is the most important module of the whole system. The module is responsible for processing and analyzing reviews, and returning the final results to the web interface. The processing flow of the module is shown in Figure 24.

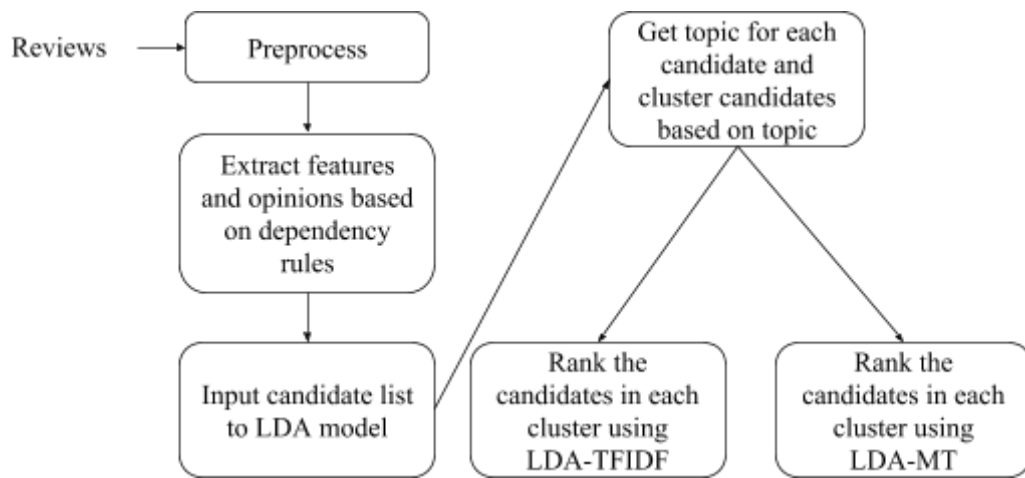


Figure 24. Processing flow of the keyphrase extraction system

For each cluster, the candidate keyphrase that has the highest score will be returned.

5.3 Web Interface Module

This module is a simple website, created mainly to facilitate users to query for the product, as well as display the results. The processing flow is shown in Figure 25.

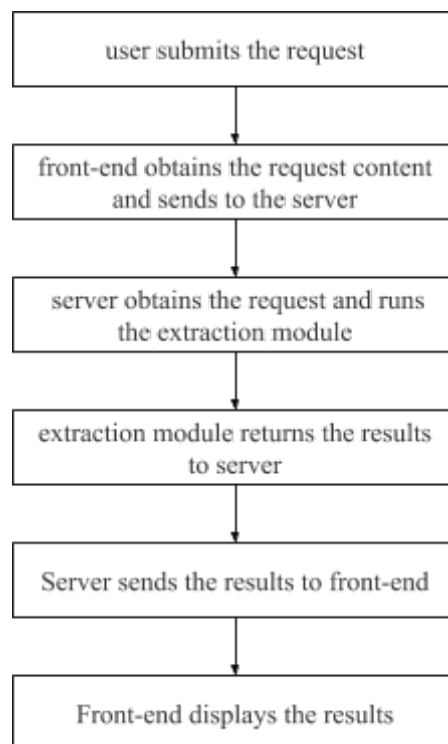


Figure 25. Processing flow of the web interface module

Figure 26 shows the interface of the website.

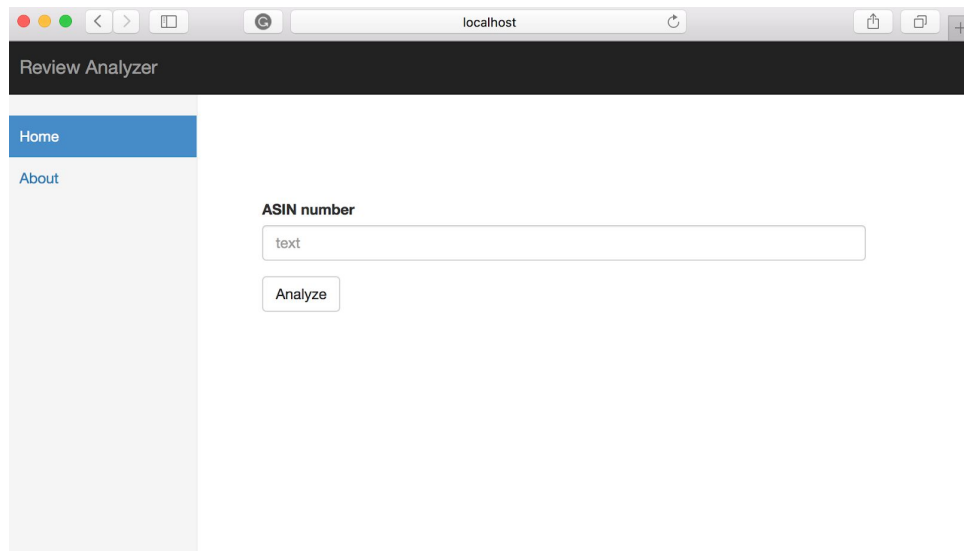


Figure 26. Website interface

6. Conclusion

With the rapid development of the internet, the total amount of web information is exploding. Therefore, an efficient and accurate information processing tool is strongly needed. Most of the online data are stored as text, attracting a lot of research focusing on natural language processing. E-business websites such as Amazon can produce thousands of reviews every minute, and these reviews contain abundant information so that they have a very high research value, which has made review mining become one of the most popular research topics in recent years.

In this thesis, reviews from Amazon, one of the world's largest e-commerce companies, are selected as the research domain. This thesis first implements a Python web crawler to crawl the target pages. By analyzing the DOM structure of Amazon review page, the crawler is designed only to extract the review related information, and then store the reviews into a local Mongo database.

After reviewing different studies about review mining, this thesis proposes and implements an automatic keyphrase extraction system. The system completes the task in four steps. Firstly, the reviews are preprocessed, including POS tagging, lemmatization, cleaning, and segmentation. Secondly, the features and opinions are extracted. This thesis adopts a similar method as the one described in Feng's [45] paper, which uses several predefined dependency rules to extract the feature and opinion words. This thesis defines three different extraction rules by observing Amazon camera reviews. More specifically, the review sentences are passed into the Spacy Dependency parser to get the dependency relations between each word, and if the relation conforms to the rules, the related words will be extracted. The extracted words will then be added into the candidate list in the form of <feature, modifier, opinion>.

Thirdly, since this thesis aims to extract semantically independent keyphrases, it is necessary to further process the candidates. One way to solve the problem is by clustering the candidates. The concept is that similar candidates will be grouped, so that each cluster can provide a semantically independent keyphrase. However, simple clustering methods such as string matching have certain limitations, for example, "cost" and "price" refer to the same object, but they cannot be grouped together according to string matching. Therefore, this thesis chooses an advanced text clustering method, which is the LDA model, a prevalent generative model for documents applicable to text clustering.

The LDA model is a three-layer Bayesian model from document to word, and the middle layer is a hidden topic layer, which describes the latent semantic information

of the document. That is to say, the LDA model can recognize hidden semantics of the text. In addition, it has already been confirmed that the topics can be regarded as product features [53]. Therefore, the LDA model has a lot advantages in feature clustering.

Finally, after getting several clusters from the LDA model, this thesis applies two ranking algorithms to sort the candidates in each clusters. The candidates with the highest score in each cluster will be chosen as the representative keyphrases forming the final keyphrase list. The algorithms are LDA-TFIDF and LDA-MT respectively. The concept of the LDA-TFIDF algorithm is to treat each keyphrase cluster as a document. Words that constantly repeat in the same document, while rarely appear in other documents have a higher probability of becoming the keywords in the current document (cluster). For each keyphrase in the cluster, the score is the sum of the TFIDF scores of the words it contains, and the keyphrase with the highest score becomes the representative keyphrase which will be displayed on the system page. Differently, the concept of the LDA-MT algorithm is, for each keyphrase in the cluster, if the words it contains have a greater contribution to the topic of the cluster, then the keyphrase is considered to have stronger ability to represent the current cluster. Similarly, the keyphrase with the highest score will be selected as the representative keyphrase.

In order to prove the validity of the automatic keyphrase extraction system, the results of the two algorithms are analyzed and compared against a manual extraction result in a preliminary small scale study. The experimental results show that both algorithms achieve relatively high accuracy, in which the performance of LDA-TFIDF is better than LDA-MT when the number of clusters is more than 10. When the number of clusters is in the middle range, that is, $5 \leq K \leq 15$, the average accuracy of both algorithms is higher than 0.5, for a system with unsupervised learning algorithms, the accuracy is already very good.

However, although the automatic keyphrase system is implemented in this thesis, there are still many deficiencies in the whole process.

Firstly, the Python crawler implemented in this thesis is not very efficient, because it is not completely automatic and it needs an input target URL to start the crawling process. In future development, the crawler can be optimized so that it can automatically identify and iterate through the review pages.

Secondly, this thesis only takes camera reviews as the analysis object. However, for other products such as clothing and food, people may use different expressions in the comments, and the proposed dependency rules may not work. In future work, a more massive corpus that consists of different kinds of product reviews can be collected

and analyzed, in order to create more comprehensive rules to improve the system accuracy.

Finally, the system is currently implemented by serial processing because the total amount of reviews is not very large. In future work, it is likely to process a huge amount of reviews, serial processing is not suitable anymore. In such situation, the system can be improved to make use of parallel processing to improve the processing speed.

References

- [1]Mayer-Schönberger, V. (2013). "Big data : a revolution that will transform how we live, work, and think".
- [2]VK Jain, V. (2017). "Big Data and Hadoop".
- [3]David Sayce. [online] David Sayce. Available at: <https://www.dsayce.com/social-media/tweets-day/> [Accessed 20 Sep. 2017].
- [4]Saleh, K. (2015.). The Importance Of Online Customer Reviews. Retrieved November 6, 2017, from <https://www.invespcro.com/blog/the-importance-of-online-customer-reviews-infographic/>
- [5]Bamane, P. (2016). A Study of Amazon User Review Data using Visualization (Rep.).
- [6]Internetlivestats.com. (2017). Total number of Websites - Internet Live Stats. [online] Available at: <http://www.internetlivestats.com/total-number-of-websites/> [Accessed 2 Oct. 2017].
- [7]Liu, B. (2007). Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. 1st ed. Springer, p.7.
- [8]Etzioni, O. (1996). The World-Wide Web: quagmire or gold mine?. *Communications of the ACM*, 39(11), pp.65-68.
- [9]Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp.168-177.
- [10]Broder, A. and Glassman, S, etc. (1997). Syntactic clustering of the Web. *Computer Networks and ISDN Systems*, 29(8-13), pp.1157-1166.
- [11]Pak, A. and Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. Conference: *Proceedings of the International Conference on Language Resources and Evaluation*, LREC 2010, pp.17-23.
- [12]Kim, S. and Hovy, E. (2006). Extracting opinions, opinion holders, and topics expressed in online news media text. *Proceedings of the Workshop on Sentiment and Subjectivity in Text*, pp.1-8.

- [13]Tumasjan, A. and Sprenger, T. etc. (2010). Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. *Fourth International AAAI Conference on Weblogs and Social Media*, ICWSM 2010, pp.23-26.
- [14]Sclaroff, S., Tayche, L. and La Cascia, M. (1997). ImageRover: a content-based image browser for the World Wide Web. *Content-Based Access of Image and Video Libraries*, 1997.
- [15]Wu, Z. and Ke, Q. (2009). Bundling features for large scale partial-duplicate web image search. *Computer Vision and Pattern Recognition*, 2009.
- [16]Ghias, A. and Logan, J. etc. (1995). Query By Humming -- Musical Information Retrieval in an Audio Database. *ACM Multimedia 95 - Electronic Proceedings*.
- [17]Pang, B. and Lee, L. (2002). Thumbs up?: sentiment classification using machine learning techniques. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, 10, pp.79-86.
- [18]Saleh, K. (2017). The Importance Of Online Customer Reviews. [online] <https://www.invespcro.com/>. Available at:<https://www.invespcro.com/blog/the-importance-of-online-customer-reviews-infographic/> [Accessed 26 Sep. 2017].
- [19]Popescu, A. M., Nguyen, B., & Etzioni, O. (2005). OPINE: extracting product features and opinions from reviews. *Proceedings of HLT/EMNLP on Interactive Demonstrations*, pp. 32-33
- [20]McAuley, J. and Leskovec, J. (2013). Hidden factors and hidden topics: understanding rating dimensions with review text. *Proceedings of the 7th ACM conference on Recommender systems*, pp.165-172.
- [21]Ganesan, K. and Zhai, C. (2011). Opinion-Based Entity Ranking. *Information Retrieval*, 2011.
- [22]Ling, G., Lyu, M., & King, I. (2014). Ratings meet reviews, a combined approach to recommend. *Proceedings of the 8th ACM Conference on Recommender systems*, 105-112.
- [23]Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., & Ma, S. (2014). Explicit Factor Models for Explainable Recommendation based on Phrase-level Sentiment

Analysis. *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, 83-92.

[24]Kasper, W. and Vela, M. (2011). Sentiment Analysis for Hotel Reviews. *Proceedings of the Computational Linguistics-Applications Conference*, 2011, pp.45-52.

[25]Owsley, S., Sood, S. and Hammond, K. (2006). Domain Specific Affective Classification of Documents.

[26]Myllymäki, J. (2001). Effective Web data extraction with standard XML technologies. *Proceedings of the 10th international conference on World Wide Web*, pp.689-696.

[27]Chau, D., Pandit, S., Wang, S. and Faloutsos, C. (2007). Parallel Crawling for Online Social Networks. *Proceedings of the 16th international conference on World Wide Web*, pp.1283-1284.

[28]Cheng, M. (2011). Web Data Mining Based on Cloud-computing. *Computer Science*, 2011

[29]Yi, J. and Niblack, W. (2005). Sentiment mining in WebFountain. *Proceedings of the 21st International Conference on Data Engineering*, pp.1073-1083.

[30]Zhuang, L., Jing, F. and Zhu, X. (2006). Movie Review Mining and Summarization. *Proceedings of the 15th ACM international conference on Information and knowledge management*, pp.43-50.

[31]Blair-goldensohn, S. and Hannan, K. etc. (2008). Building a sentiment summarizer for local service reviews. *Proceedings of the WWW2008 Workshop: NLP in the Information Explosion Era* (NLPIX 2008).

[32]Yao, T. and Nie, Q. (2006). An Opinion Mining System for Chinese Automobile Reviews.

[33]Kobayashi, N., Inui, K., Matsumoto, Y., Tateishi, K., & Fukushima, T. (2005). Collecting Evaluative Expressions for Opinion Extraction. *Natural Language Processing – IJCNLP 2004 Lecture Notes in Computer Science*, pp. 596-605

[34]Liu, B., Hu, M., & Cheng, J. (2005). Opinion Observer: Analyzing and Comparing Opinions on the Web. *Proceedings of the 14th international conference on World Wide Web - WWW 05*.

- [35]Agrawal, R. and Srikant, R. 1994. Fast algorithm for mining association rules. VLDB'94.
- [36]Kessler, J. S., & Nicolov, N. (2009). Targeting Sentiment Expressions through Supervised Ranking of Linguistic Configurations. *Proceedings of the Third International ICWSM Conference*.
- [37]Bloom, K.; Garg, N.; and Argamon, S. (2007). Extracting Ap- praisal Expressions. In NAACL-HTL.
- [38]Liu, B., Hsu, W., Ma, Y. 1998. Integrating Classification and Association Rule Mining. KDD'98, 1998.
- [39]O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. (2005). Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134.
- [40]Hai Z., Chang K., Kim J. (2011) Implicit Feature Identification via Co-occurrence Association Rule Mining. In: Gelbukh A.F. (eds) *Computational Linguistics and Intelligent Text Processing*. CICLing 2011. Lecture Notes in Computer Science, vol 6608. Springer, Berlin, Heidelberg
- [41]Qiu , G., Zheng, M., Zhang, H., Zhu, J., Bu, J., Chen, C., & Hang, H. (2011). Implicit product feature extraction through regularized topic modeling. *Journal of Zhejiang University(Engineering Science)*.
- [42]Ku, L., Liang, Y., & Chen, H. (2006). Opinion Extraction, Summarization and Tracking in News and Blog Corpora.
- [43]Mei, J., Zhu, Y. Gao, Y. and Yin, H.. tong2yi4ci2ci2lin2. Shanghai Dictionary Press. 1982.
- [44]Hu, M., & Liu, B. (2006). Opinion feature extraction using class sequential rules. *Computational Approaches to Analyzing Weblogs*, Papers from the 2006 AAAI Spring Symposium, Technical Report SS-06-03, Stanford, California, USA.
- [45]Feng, S., Zhang, M., Zhang, Y., & Deng, Z. (2010). Recommended or Not Recommended? Review Classification through Opinion Extraction. *Advances in Web Technologies and Applications*, Proceedings of the 12th Asia-Pacific Web Conference, APWeb 2010, Busan, Korea, 6-8.

- [46]Yi, J., Nasukawa, T., Bunesco, R., & Niblack, W. (2003). Sentiment Analyzer: Extracting sentiments about a given topic using natural language processing techniques. *The Third IEEE International Conference on Data Mining*.
- [47]Miao, Q., Li, Q., & Zeng, D. (2010). Mining Fine Grained Opinions by Using Probabilistic Models and Domain Knowledge. *Web Intelligence and Intelligent Agent Technology (WI-IAT)*.
- [48]Mishra, R. (2010). FEROM (Feature extraction and refinement) using genetic algorithm. *Applied and Theoretical Computing and Communication Technology (iCATccT)*.
- [49]Huang, A., Milne, D., Frank, E., & Witten, I. H. (2009). Clustering Documents Using a Wikipedia-Based Concept Representation. *Advances in Knowledge Discovery and Data Mining Lecture Notes in Computer Science*, 628-636.
doi:10.1007/978-3-642-01307-2_62
- [50]Banerjee, S., Ramanathan, K., & Gupta, A. (2007). Clustering short texts using wikipedia. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR 07*.
doi:10.1145/1277741.1277909
- [51]Hu, X., Sun, N., Zhang, C., & Chua, T. (2009). Exploiting internal and external semantics for the clustering of short texts using world knowledge. *Proceeding of the 18th ACM conference on Information and knowledge management - CIKM 09*.
doi:10.1145/1645953.1646071
- [52]Petersen, H., & Poon, J. (2011). Enhancing Short Text Clustering with Small External Repositories. *Proceedings of the Ninth Australasian Data Mining Conference*, 121, 79-90.
- [53]Wang, W., & Meng, C. (2011). Opinion object extraction based on the syntax analysis and dependency analysis. *Computer Systems & Applications*, 52-57.
- [54]S. Deerwester, S. Dumais, T. Landauer, G. Furnas, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391-407, 1990.
- [55]Hofmann, T. (1999). Probabilistic latent semantic indexing. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 50-57.

- [56]Blei, D., Ng, A., & Jordan, M. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research*, 993-1022.
- [57]Zhang, C., Sun, J., & Ding, Y. (2011). Topic mining for microblog based on MB-LDA model. *Journal of computer research and development*.
- [58]Tang, X., & Xiao, L. (2014). Research of micro-blog topics mining based on granularity. *Journal of the China Society for Scientific and Technical Information*, 623-632.
- [59]Lu, Y., Zhai, C., & Sundaresan, N. (2009). Rated aspect summarization of short comments. *Proceedings of the 18th international conference on World wide web - WWW 09*. doi:10.1145/1526709.1526728
- [60]Titov, I., & Mcdonald, R. (2008). Modeling online reviews with multi-grain topic models. *Proceeding of the 17th international conference on World Wide Web - WWW 08*. doi:10.1145/1367497.1367513
- [61]Jo, Y., & Oh, A. H. (2011). Aspect and sentiment unification model for online review analysis. *Proceedings of the fourth ACM international conference on Web search and data mining - WSDM 11*. doi:10.1145/1935826.1935932
- [62]Guo, H., Zhu, H., Guo, Z., Zhang, X., & Su, Z. (2009). Product feature categorization with multilevel latent semantic association. *Proceeding of the 18th ACM conference on Information and knowledge management - CIKM 09*. doi:10.1145/1645953.1646091
- [63]Tu, D., & Chen, L. (2013). Multi-way hierarchical clustering based concept taxonomy construction for product reviews. *Journal of computer research and development*, 208-215.
- [64]Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management: an International Journal*, 24(5), 513-523.
- [65]Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010). Automatic Keyword Extraction from Individual Documents. In book: *Text Mining: Applications and Theory*, 1-20.
- [66]Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing Order into Text.

- [67]Litvak, M., & Last, M. (2008). Graph-based keyword extraction for single-document summarization. *Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization*,17-24.
- [68]Tsatsaronis, G., Varlamis, I., & Nørvåg, K. (2010). SemanticRank: ranking keywords and sentences using semantic graphs. *Proceedings of the 23rd International Conference on Computational Linguistics* , 1074-1082.
- [69]Dave, K., Lawrence, S., & Pennock, D. M. (2003). Mining the peanut gallery. *Proceedings of the twelfth international conference on World Wide Web - WWW 03*. doi:10.1145/775152.775226
- [70]Gamon, M., Aue, A., Corston-Oliver, S., & Ringger, E. (2005). Pulse: Mining Customer Opinions from Free Text. *Lecture Notes in Computer Science Advances in Intelligent Data Analysis VI*, 121-132. doi:10.1007/11552253_12
- [71]Baoguo, L. (2016). News Review Topic Mining Based on Clustering and LDA. *Master thesis collection of Wuhan Textile University, 2016*.
- [72]Woolf, M. (2014). A Statistical Analysis of 1.2 Million Amazon Reviews. Retrieved November 27, 2017, from <http://minimaxir.com/2014/06/reviewing-reviews/>
- [73]Liu, J., Cao, Y., Lin, C., & Zhou, M. (2007). Low-Quality Product Review Detection in Opinion Summarization. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 28-30.
- [74]Atefeh, H., Mohammad, T., Naomie, S., & Zahra, H. (2015). Detection of review spam: A survey. *Expert Systems with Applications, Volume 42, Issue 7, 2015, Pages 3634-3642, ISSN 0957-4174*.
- [75]Ding, X., Liu, B., & Yu, P. S. (2008). A holistic lexicon-based approach to opinion mining. *Proceedings of the international conference on Web search and web data mining - WSDM 08*. doi:10.1145/1341531.1341561
- [76]Bishop, C. M. (2016). Pattern Recognition and machine learning. S.l.: Springer-Verlag New York.
- [77]Blei, D. M. (2012). Probabilistic Topic Models. *Commun. ACM*, 55, 77--84. doi:10.1145/2133806.2133826

[78]Stein, M., Griffiths, T. (2007). Probabilistic topic models. *Latent Semantic Analysis: A Road to Meaning*.

[79]Hoffman, M., Bach, F., & Blei, D. (2010). Online Learning for Latent Dirichlet Allocation. *Proceedings of the 23rd International Conference on Neural Information Processing Systems*, 856-864.

[80]Heinrich, Gregor. (2005). Parameter Estimation for Text Analysis.

[81]Turney P D. Learning algorithms for keyphrase extraction [J]. *Information retrieval*, 2000, 2(4): 303-336.